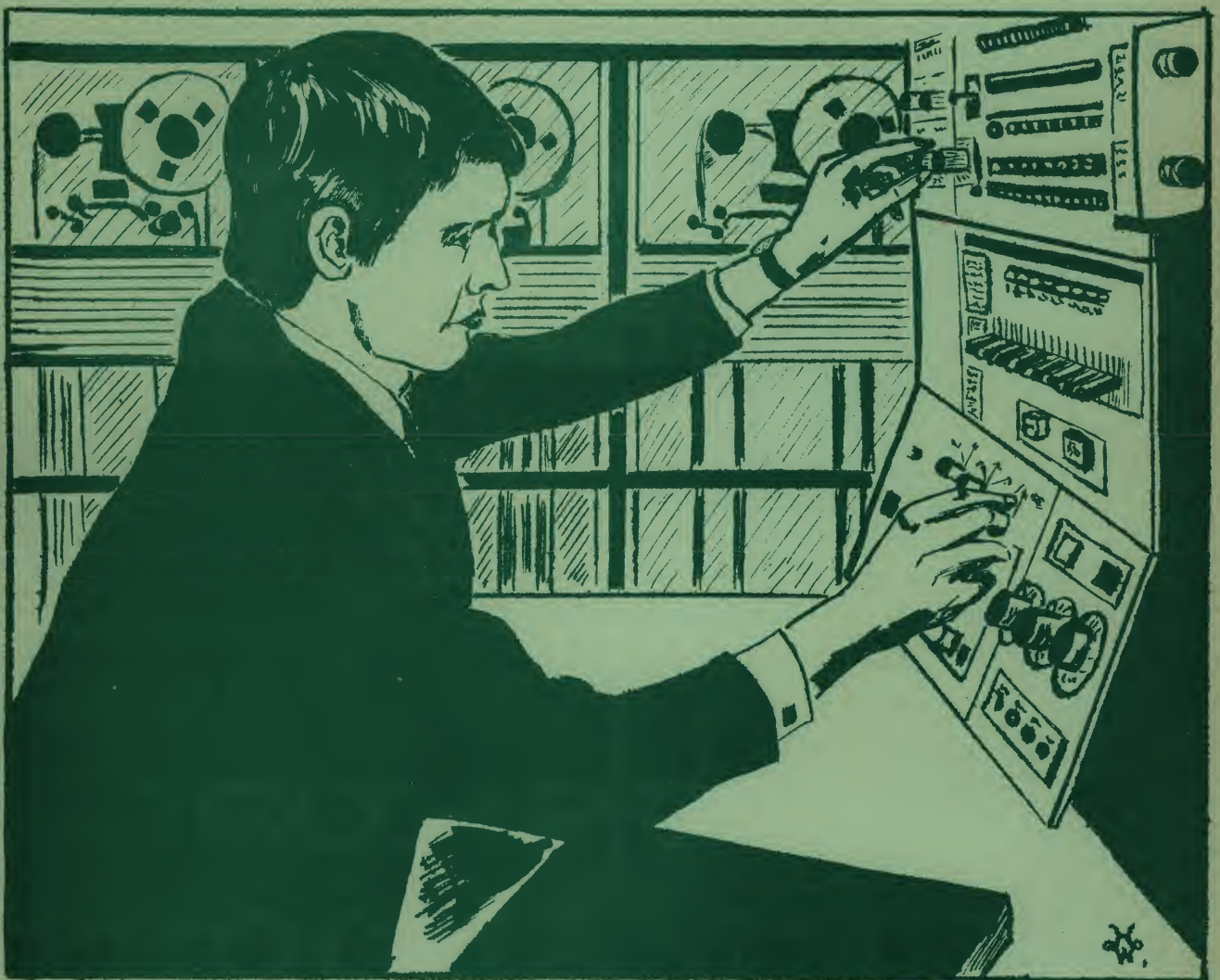


# COMPUTER PROGRAMMEUR



Les 2



### 1. INLEIDING.

In dit lesdeel zal aandacht worden besteed aan de in- en uitvoer van informatie, alsmede aan de tot stand koming van een programma.

Andere woorden voor invoer en uitvoer zijn INPUT en OUTPUT.

Wij zullen echter eerst een zeer beknopte herhaling geven van het voorgaande lesdeel.

### HERHALING

In een vorige les hebben we er reeds op gewezen dat de werking van de digitale rekenmachine berust op het principe van het binair talstelsel. M.a.w. het rekenen gebeurt in het tweetalig stelsel, waarbij alle informatie in het geheugen d.m.v. bits wordt opgeslagen.

Gesproken is er ook over het kerntjesgeheugen. Zo'n kerntje kon daarbij links- of rechtsom gemagnetiseerd zijn. Een bit wordt dan voorgesteld door een kerntje.

Een Sera-woord hebben we afgesproken, bevat 48 bits. Een groep van bits noemen we een woord.

Zo'n woord kan echter slechts één soort informatie bevatten, n.l. een combinatie van enen en nullen. De inhoud van een Sera-woord kan echter een verschillende betekenis hebben, zoals een *getal*, een *instructie*, of een *tekst*. Dit is afhankelijk van de betekenis die men aan de bits of groepen van bits toekent.

### HET PROGRAMMA

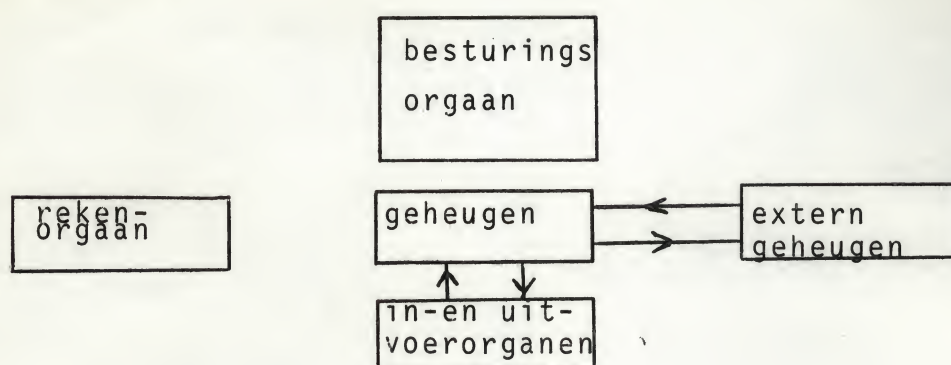
Een programma bestaat uit een reeks instructies door de programmeurs geschreven, teneinde door deze instructies los te laten op een computer, een vooraf vastgesteld doel te bereiken.

Vooraf moeten deze instructies in achtereenvolgende woorden in het geheugen worden vastgelegd. Pas daarna kunnen ze worden uitgevoerd.

Het grote probleem is nu hoe krijgt men het programma in het geheugen van de machine? Bovendien levert het programma resultaten op die eerst in het geheugen worden opgeborgen om pas daarna naar een uitvoereenheid te worden gebracht. Ook hier zullen dus middelen moeten zijn om dit te kunnen verwezenlijken.

In de nu volgende afbeelding ziet U nogmaals een schema van de opbouw van een rekenmachine. Wij hebben hier bewust de verbindingslijnen tussen geheugen- en reken- en besturingsorgaan weggelaten. In deze les interesseert ons slechts de verbinding tussen het geheugen en de in- en uitvoerorganen.





## 2. IN- EN UITVOER VAN INFORMATIE

2. IN- EN UITVOER VAN INFORMATIE  
Het zal U ongetwijfeld duidelijk zijn dat de in- en uitvoer van gegevens in de machine zelf een gecompliceerd geheel is. Dit vindt voor een groot deel zijn oorzaak in het feit dat gegevens door de mens in normaal geschreven schrift door de machine slechts verwerkt kan worden indien het omgezet is in een binair systeem. Omgekeerd zullen gegevens uit de machine vanuit een binair systeem omgezet moeten worden in een voor de mens leesbaar gegeven.

### Voorbeeld

Een stukje programma dat drie getallen, die op de adressen 300, 301 en 302 in het geheugen, optelt en het resultaat weer zal opbergen in het geheugenadres 303, zal er door een programmeur geschreven, bijvoorbeeld als volgt uitzien:

100	HPA	300
101	OPA	301
102	OPA	302
103	BPA	303

Dit stukje programma kan nu in het geheugen worden gebracht. We nemen daarbij aan dat de mnemotechnische opdrachten HPA, OPA en BPA overeenkomen met resp. de interne opdrachten 20, 30 en 34. De 48 bits van de geheugenwoorden op de adressen 100, 101, 102 en 103 zullen dan de volgende vorm hebben.

[illegible]

Hoewel de techniek zich op dit terrein snel evolueert is het thans nog niet mogelijk een op papier geschreven programma rechtstreeks door de machine te laten "LEZEN". Aan de computer zal n.l. een apparaat gekoppeld zijn dat in staat is symbolen af te tasten en de "gelezen" informatie door te geven aan de rekenmachine. De symbolen die de machine aftast zijn vastgelegd op een "Informatie-drager" een begrip waarover we reeds eerder spraken.



H. Wenteler

Het zal U duidelijk zijn dat de machine het gemakkelijkste informatie kan lezen die in binaire vorm op de informatiedrager is vastgelegd.

Als informatiedrager kent de machine vele vormen. Zo kan men bijv. papieren ponsband of kartonnen ponskaarten gebruiken waarin de informatie d.m.v. een gaatjes codering wordt vastgelegd. Indien een gaatje op een bepaalde positie is vastgelegd kan dit dan een bitwaarde vertegenwoordigen die gelijk is aan 1. Is er op een bepaalde plaats geen gaatje aangebracht, dan zal de bitwaarde 0 zijn. De kunst is nu dat de machine die als leesapparaat dienst doet, in staat moet zijn onderscheid te maken tussen een ponsgaatje en een niet geponste plek.

Het enigste probleem wat er speelt is het omzetten van het normaal op papier geschreven schrift in een combinatie van ponsgaatjes. Dit gebeurt echter op de ponskamer, waar handige ponsstypistes het geschreven schrift inponsen op een ponsmachine. Als uitvoer kan dan een ponskaart of ponsband ontstaan die de gewenste ponscodes bevat. M.a.w. het omzetten van normaal schrift in bepaalde ponscodes gebeurt eigenlijk intern in de daarvoor bestemde ponsmachine.

Het doel van het lezen is het *INBRENGEN VAN INFORMATIE IN HET GEHEUGEN*.

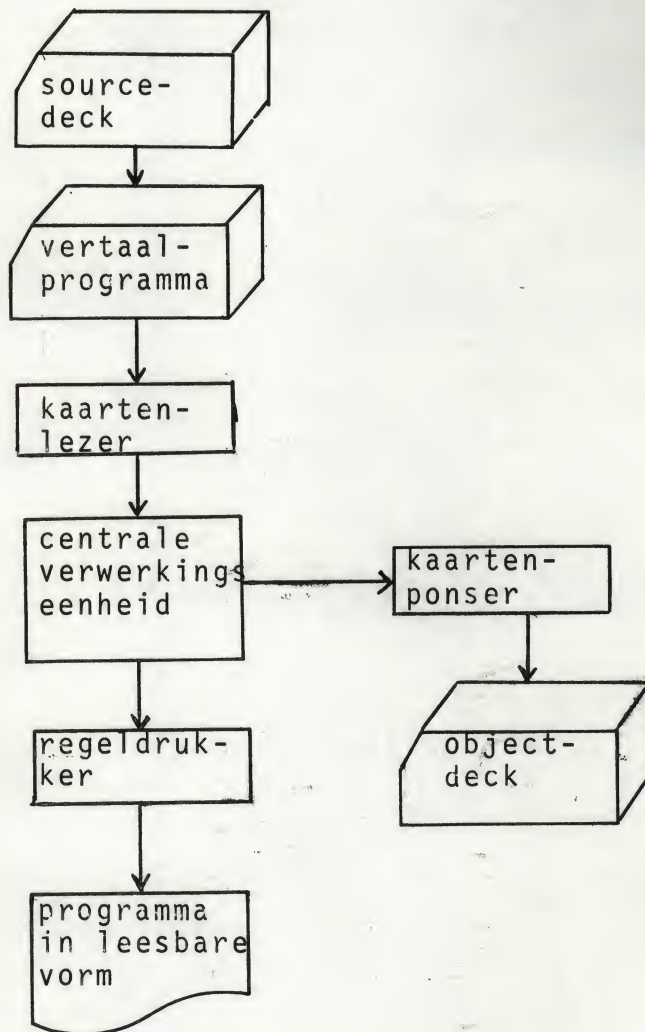
Indien we "uitvoer" van de computer wensen in normaal leesbaar schrift, zullen de coderingen intern weer worden omgezet. We zullen er in dit deel van de cursus nog niet te diep op ingaan. Voorlopig kunt U volstaan met te onthouden dat de ponskaart één van de meest gebruikte invoervormen is. Terwijl als uitvoer het meest gebruik wordt gemaakt van de *SNELDRUKKER*; Dit is de machine die informatie in voor de mens normaal leesbaar schrift kan afdrukken.

#### EEN PROGRAMMA KOMT IN HET COMPUTERGEHEUGEN

Aan de hand van onderstaand schema geven wij een beknopte beschrijving over het in het geheugen brengen van een computerprogramma. Het heeft immers weinig zin om over in- en uitvoer van informatie te praten als we nog niet gesproken hebben over inbrengen van een programma.

Nadat het door de programmeur geschreven programma in *symbolische* taal is geschreven wordt het door de ponsstypiste overgebracht in ponskaarten. Het aldus ontstane pakket ponskaarten in symbolische taal noemt men wel het *SOURCE-DECK*. Tesamen met het *vertaalprogramma* wordt het nu in de kaartenlezer ingevoerd. Het vertaalprogramma heeft tot taak de instructies die zich in de ponskaart in ponscode bevinden, om te zetten in voor de machine begrijpelijke instructies. M.a.w. het programma wordt vanuit een *symbolische* taal omgezet in een *MACHINE TAAL*.



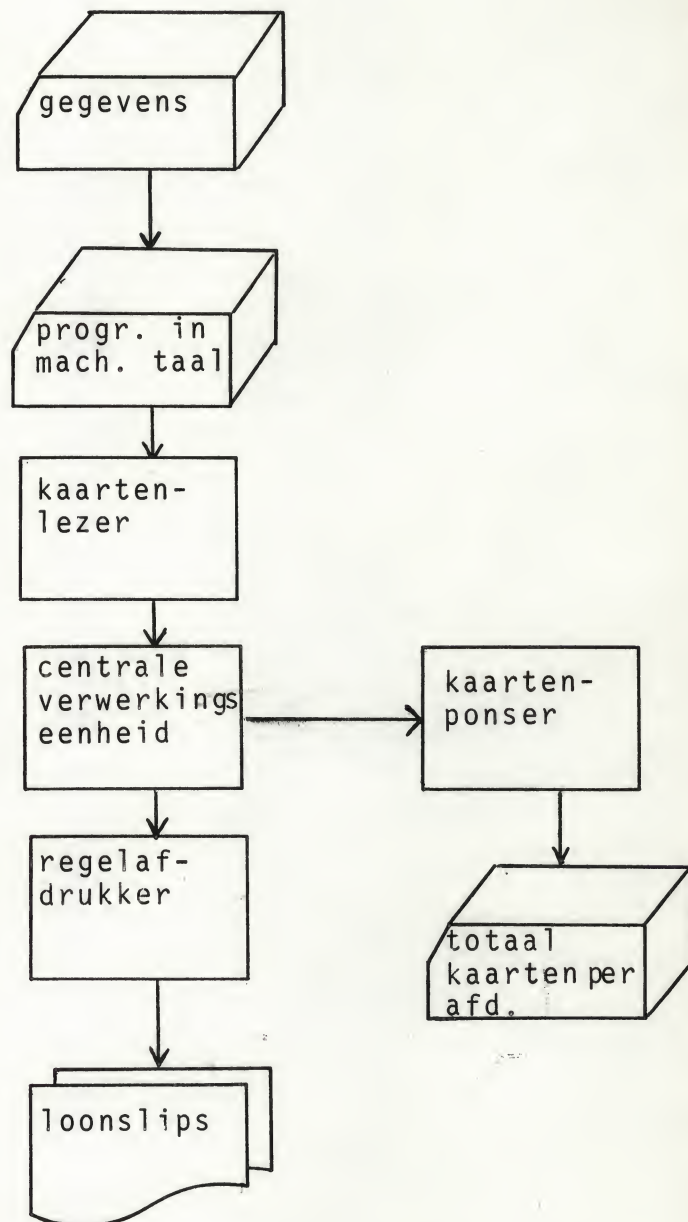


Het vertalen van een programma geschreven in symbolische taal.

Dit "vertalen" gebeurt in de Centrale Verwerkings Eenheid van de computer. Door de sneldrukker in te schakelen is het ons mogelijk om een afdruk te krijgen van het programma. Tevens is het via de sneldrukker mogelijk gegevens te krijgen over fouten in het programma, gebruikte geheugenruimte, enz. Indien wij het programma willen archiveren voor bijv. gebruik op een later tijdstip, hebben wij de mogelijkheid de "vertaling" van het programma uit te laten ponsen. Dit programma wat men ook wel *OBJECT-DECK* noemt is in machine-taal, zodat bij een later gebruik de "vertaalfase" overgeslagen kan worden.



Het ontstane object-deck is nu het uiteindelijke programma dat tesamen met de te verwerken informatie wordt ingevoerd. In bijgaand schema ziet U een zeer beknopt voorbeeld van een toepassing.



De uitvoering van een programma.

Bijvoorbeeld van een loonberekening.



IN- EN UITVOER

Ir. P.A. Tas

Een publikatie van de Stichting Het Nederlands Studiecentrum  
voor Informatica, Amsterdam.

Copyright © Studiecentrum voor Informatica, 1971.  
Niets uit deze uitgave mag worden vermenigvuldigd  
en/of openbaar gemaakt door middel van druk, foto-  
kopie, mikrofilm of op welke andere wijze dan ook  
zonder voorafgaande schriftelijke toestemming van  
de uitgever.



## 1. INLEIDING

Het is met de huidige stand van de techniek nog niet mogelijk een op papier geschreven programma direkt door de machine te laten lezen. De symbolen letters, cijfers en leestekens, zoals die op deze bladzijde gedrukt staan, zijn voor haar onbegrijpelijk.

Het is daarom noodzakelijk gebruik te maken van een tolk die de voor ons leesbare symbolen omzet in de voor de machine leesbare symbolen. De informatie die overgebracht moet worden mag daarbij natuurlijk niet verminkt worden. Men kan het overbrengen van informatie tussen mens en machine vergelijken met de korrespondentie tussen iemand die normaal kan zien en een blinde. De geschreven tekst wordt daarbij omgezet in braille-schrift door een pons-machine. De blinde kan de braille-schrift voelen en begrijpen.

Er zijn bij de rekenmachine vele soorten informatie-dragers en daarbij behorende apparatuur mogelijk. Deze apparaten staan onder controle van het besturingsorgaan. Zeer veel gebruikte informatiedragers zijn : magneetbanden, ponskaarten en ponsband. Verder zijn er reeds apparaten die documenten kunnen lezen die bedrukt zijn met magnetische inkt symbolen of die speciale symbolen optisch kunnen onderkennen.

De SERA zal gebruik maken van magneetbanden, ponskaarten en ponsband. Dit zowel voor de in- als uitvoer. Het is bovendien mogelijk de uitvoer direkt in leesbaar schrift opgeleverd te krijgen via een regeldrukker. In dit hoofdstuk zullen de ponskaart, ponsband en regeldrukker worden behandeld met de bijbehorende instructies. Aan de magneetband zal - gezien de belangrijke toepassing en de meer ingewikkelde structuur - een geheel apart hoofdstuk, voor wat betreft de programmering, worden gewijd.

## 2. IN- EN UITVOER-APPARATUUR

### 2.1. Ponskaart

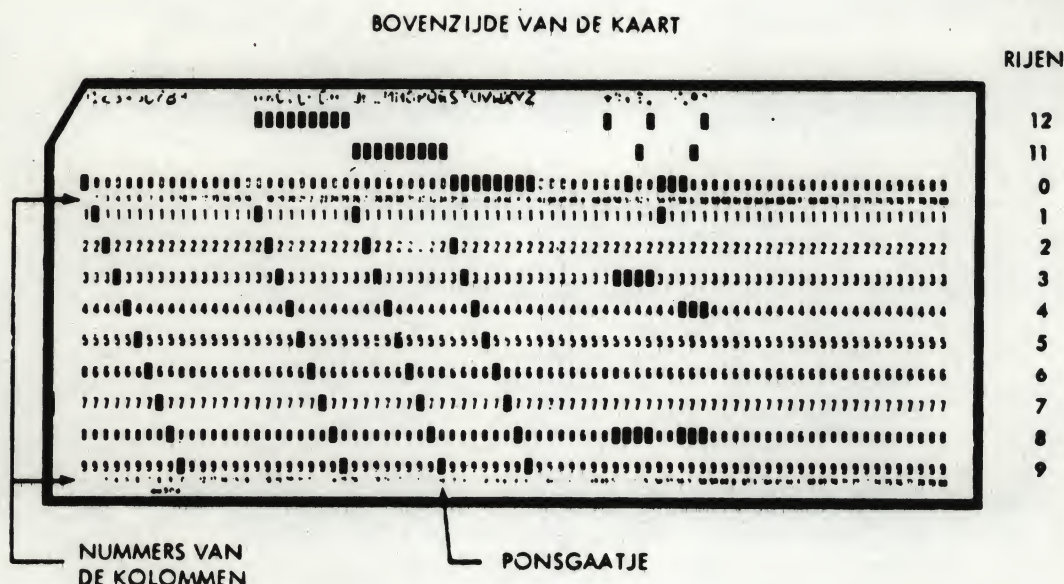
De ponskaart is een van de meest toegepaste informatie-dragers. Men is er, door de conventionele apparatuur, aan gewend en ze zijn gemakkelijk in te passen in een systeem met elektronische rekenmachine.

Een ponskaart is gemaakt van stijf papier of licht karton, waarvan de linker bovenhoek is afgesneden. De meest gebruikte kaart is verdeeld in 80 kolommen, met op elke kolom 12 posities waarin het mogelijk is een rechthoekig gat te ponsen.

De posities 0 t/m 9 staan gedrukt op de kaart, 11 en 12 niet. Deze laatste twee bevinden zich boven de 0, waarbij 12 aan de bovenkant van de kaart, 0, 11 en 12 worden zone-



posities genoemd, 1 t/m 9 numerieke posities. De bovenrand kan worden bedrukt met de tekst die geponst wordt. Op onderstaande afbeelding treft U een schets van een ponskaart aan.



De ponskaart kan in een kolom elk karakter weergeven, dat ook in de geschreven tekst staat. Die karakters zijn : letters, cijfers, leestekens en speciale tekens, zoals dollar-teken, etc. Aan één of meer ponsingen in een kolom wordt nu een speciale betekenis toegekend.

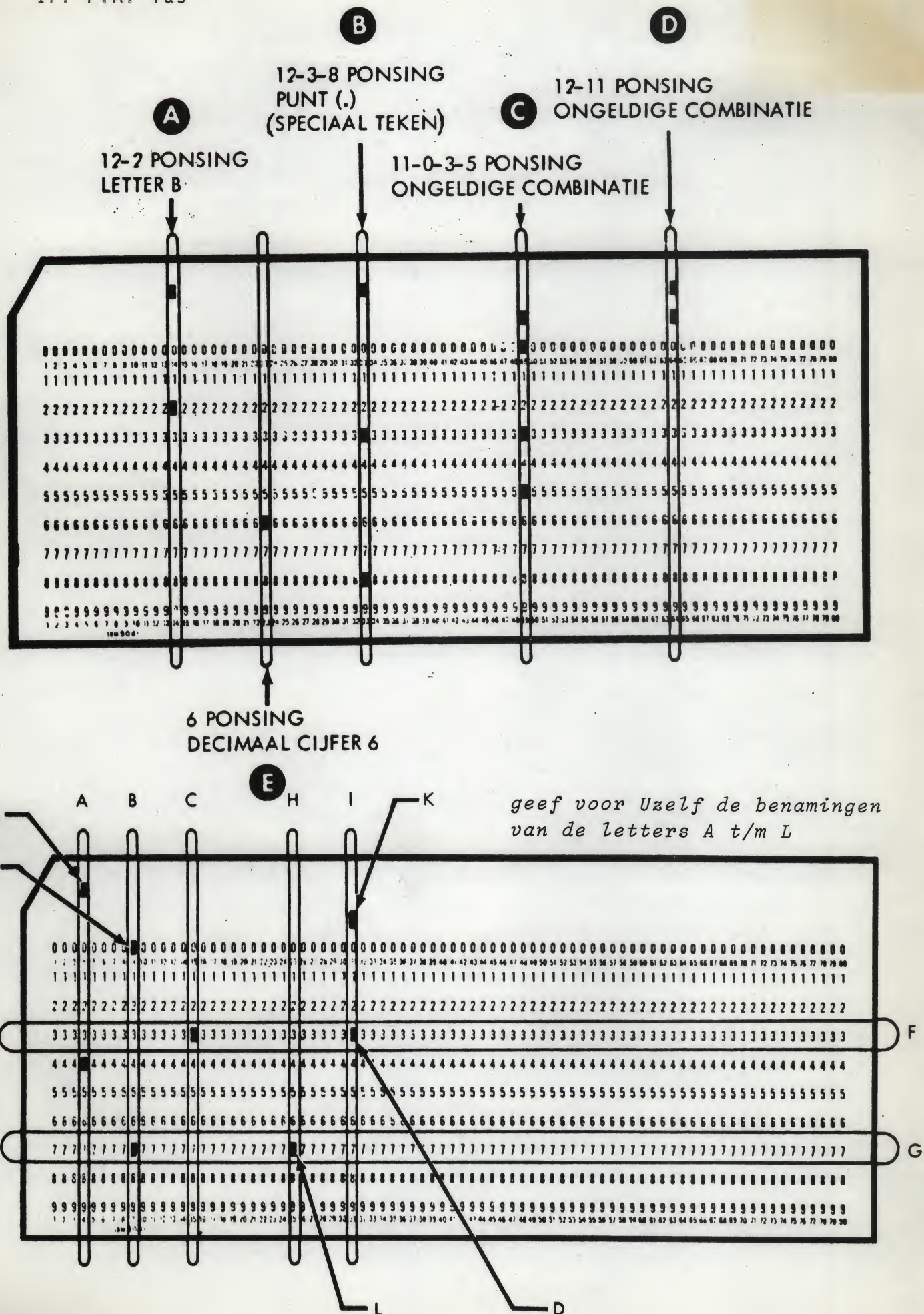
De numerieke cijfers 0 t/m 9 worden weergegeven door één ponsing en wel in de positie waar het desbetreffende getal gedrukt staat. De alfabetische karakters worden weergegeven door 2 ponsingen in eenzelfde kolom, één in de groep van de numerieke posities, één in de zone-posities. De speciale tekens worden voorgesteld door 1, 2 of 3 ponsingen in een en dezelfde kolom.

#### Voorbeeld van de SERA-kaartcode

<u>ponsingen in posities</u>	<u>betekent</u>
3	→ 3
11 en 5	→ n
12 en 8	→ h
0 en 9	→ z
11,4 en 8	→ x
0,4 en 8	→ )

Op de volgende pagina vindt U enkele voorbeelden van tekens in een ponskaart.





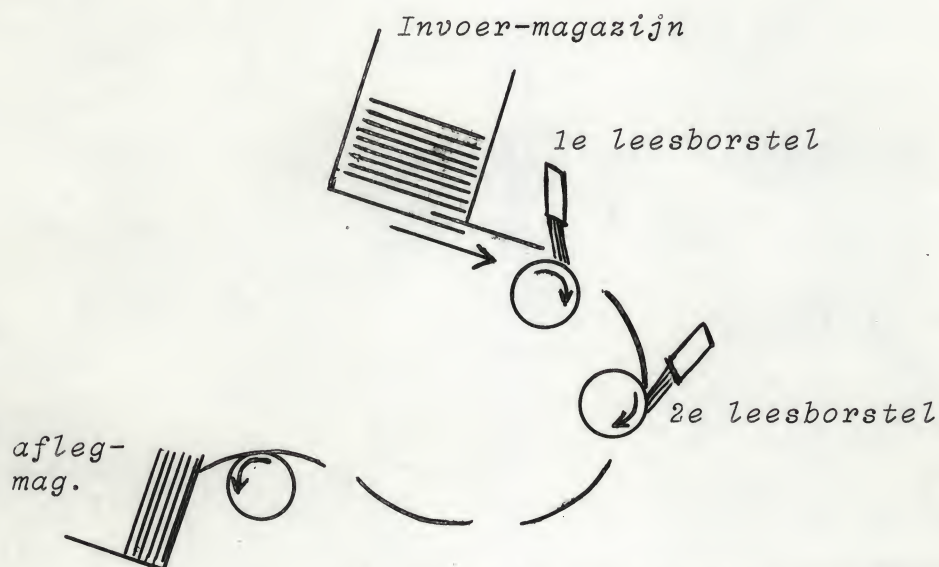


## 2.2. DE KAARTENLEZER EN PONSMACHINE

Het lezen van een kaart kan door middel van borstels of fotoelektrische cellen plaatsvinden.

De borstels ontdekken de aanwezigheid of afwezigheid van geponste gaten, doordat al of niet geleiding optreedt. Op deze wijze wordt de informatie op de kaart geconverteerd in elektrische impulsen, die naar de centrale eenheid worden gestuurd. Als de kaarten rij voor rij gelezen worden, zijn er 80 borstels aanwezig. Het is mogelijk de kaarten door middel van twee leesstations twee keer achter elkaar te lezen, de informatie in beide gevallen door te geven aan de centrale eenheid en deze daar te vergelijken. De leessnelheid varieert van 100 tot 1000 kaarten per minuut, afhankelijk van de gebruikte kaartlezer.

De kaarten worden in een leesvak gelegd. Met behulp van transportrollen worden ze langs de lezers geleid en in een aflegvak gebracht.



Schematische werking van een kaartenlezer.

De foto-elektrische lezer doet hetzelfde als het borstel-type. Slechts de methode voor het ontdekken van geponste gaten is verschillend.

Foto-elektrische cellen worden geactiveerd door licht. Indien een lichtstraal door de kaart kan vallen - omdat er een geponst gat voor ligt - wordt de bijbehorende fotocel geactiveerd.

## DE KAARTPONSER

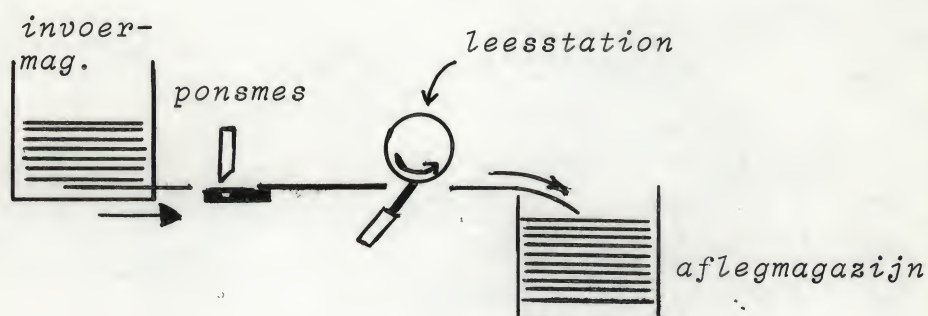
De kaartponser ontvangt vanuit de centrale eenheid de opdracht op welke posities in de kaart gaten moeten worden



geponst. Er is een vak met blanco kaarten, waarvan er automatisch telkens één aan het ponsmechanisme wordt toegevoerd. De gaten worden geponst, doordat de kaart komt te liggen op een matrijs die een gat heeft op elke mogelijke ponspositie in een rij. Bij elke mogelijke ponspositie behoort een ponsmes dat elektromagnetisch geactiveerd kan worden. Aan een ponsapparaat moeten nogal veel mechanische eisen worden gesteld. De snelheden zijn daardoor aanzienlijk lager dan die van een lezer:

100 tot 250 kaarten per min., afhankelijk van het type. Het is mogelijk de kaart, die geponst is, daarna te lezen en de gelezen informatie te vergelijken met de informatie in het geheugen, die op de kaart geponst moet worden.

Schematische werking van een kaartenponser.



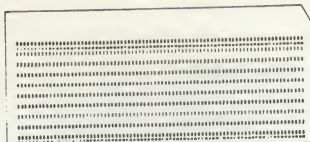
Er zijn nog andere soorten ponskaarten waarbij ronde gaten worden toegepast en die tevens minder dan 80 kolommen bezitten.

DRIEBANDPONSKAART

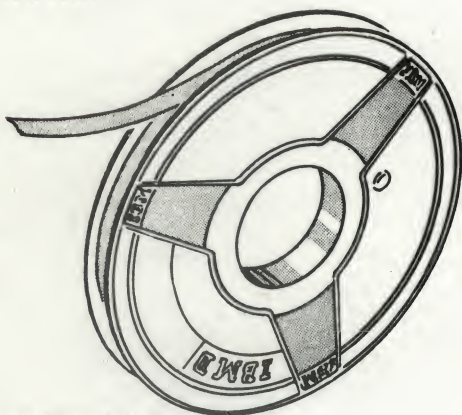
Een der grootste computerleveranciers de IBM, gebruikt voor één van haar computersystemen de driebandponskaart. Hiervan is het onderste gedeelte van de kaart het ponsgedeelte, terwijl het bovenste gedeelte van de kaart als afdruk gedeelte wordt gebruikt. Het ponsgedeelte is verdeeld in 3 gelijke delen, die elk uit 32 ponskolommen bestaan. De ponskolom bevat in tegenstelling tot de normale ponskaart 6 ponsposities. Ofschoon de kaart beduidend kleiner van afmeting is kan deze toch meer tekens bevatten nl. 96. Elke rij in het ponsgedeelte bevat 32 ponskolommen. De ponsgaatjes zijn rond i.p.v. rechthoekig zoals bij het normale model ponskaart.

2.3. PONS BAND

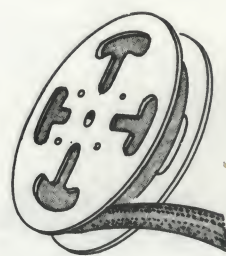
Bij vele rekencentra is de ponsband met lezer en ponser de



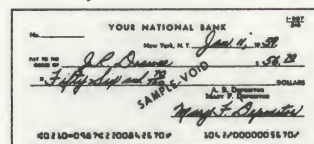
IBM KAART



MAGNETISCHE BAND



PONSBAND

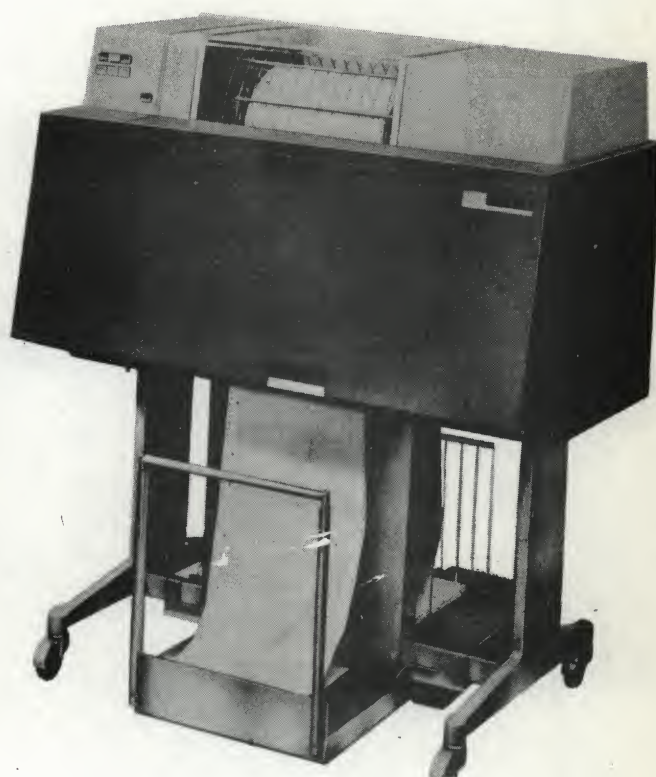


LETTERS CIJFERS EN TEKENS  
MET MAGNETISCHE INKT

*diverse invoermedia*



*kaartleeseenheid*



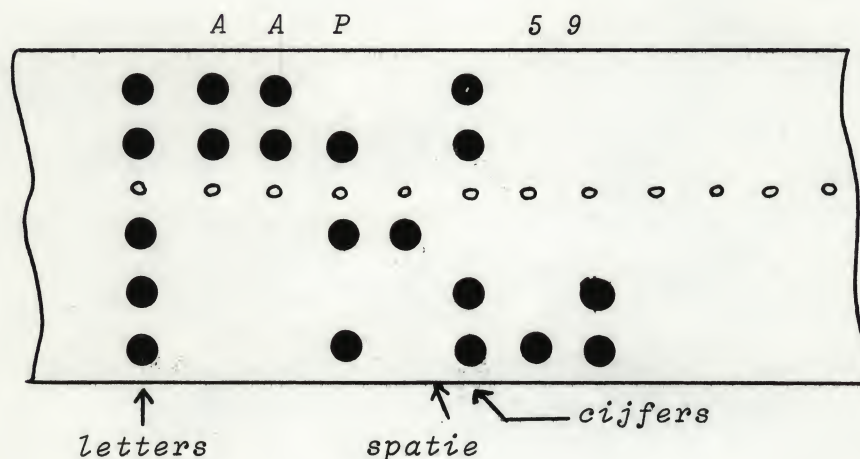
*afdrukeenheid*



belangrijkste vorm van in- en uitvoer. Meestal zal dit het geval zijn bij wetenschappelijk gebruik van de rekenmachine. Als bijkomende vorm van in- en uitvoer wordt de ponsband veel gebruikt indien de band een hoofd- of bijprodukt is van het een of ander apparaat, zoals meetapparaat, telex of een kasregister. De meest gebruikte systemen zijn die waarbij gebruik gemaakt wordt van 5, 6, 7 en 8-kanalen. Het al dan niet aanwezig zijn van een gat kan door de lezer worden gedetecteerd en doorgegeven aan de centrale eenheid.

De lezer leest telkens een kolom van de ponsband, dus bekijkt 5, 6, 7 of 8 plaatsen, naar gelang het gebruikte type ponsband, die al dan niet geponst kunnen zijn. Zo'n kolom stelt een karakter voor, dat kan bestaan uit een getal, een letter of een bijzonder teken. Indien een 5-kanaals ponsband wordt gebruikt, dan is het aantal mogelijkheden voor de ponsingen  $2^5 = 32$ .

Dit is niet genoeg om alle cijfers, letters en speciale symbolen te representeren. Daarom wordt een cijfer- en een letter-symbool ingevoerd. Wordt een letter-symbool geponst, dan zijn alle volgende kolommen letters. Dit blijft zo totdat een cijfers-symbool wordt ontmoet, waardoor de volgende kolommen gedefinieerd worden als cijfers of speciale tekens. Bijgaande figuur toont een stuk 5-kanaals ponsband, geponst in de telex-kode.



Een donker cirkeltje betekent een ponsing. Tussen het tweede en derde kanaal van boven af geteld bevindt zich een kanaal met kleine ponsingen.

#### 2.4. PONS BAND LEES- EN PONS APPARATUUR

Het konstateren van een al dan niet geponst gat geschiedt bij



de moderne lezers in het algemeen op de volgende manieren:

1. met fotocellen. Een kolom wordt van boven met licht beschienen. Aan de andere kant van het papier bevinden zich een aantal fotocellen. Op een positie waar een gat aanwezig is, valt er licht doorheen en een bijbehorende fotocel wordt geactiveerd, waardoor een bit gelijk 1 wordt gereproduceerd. Op een punt waar geen gat is, valt er geen licht door en de bijbehorende fotocel wordt dus niet geactiveerd en het bit wordt gelijk aan 0.
2. Door verandering van het dielectricum. Het principe is hier, dat de lezer bestaat uit een aantal condensatoren, waarvan tussen de beide platen dielectricum wordt gevarieerd, tengevolge van de doorgetrokken ponsband.

#### HET PONSEN

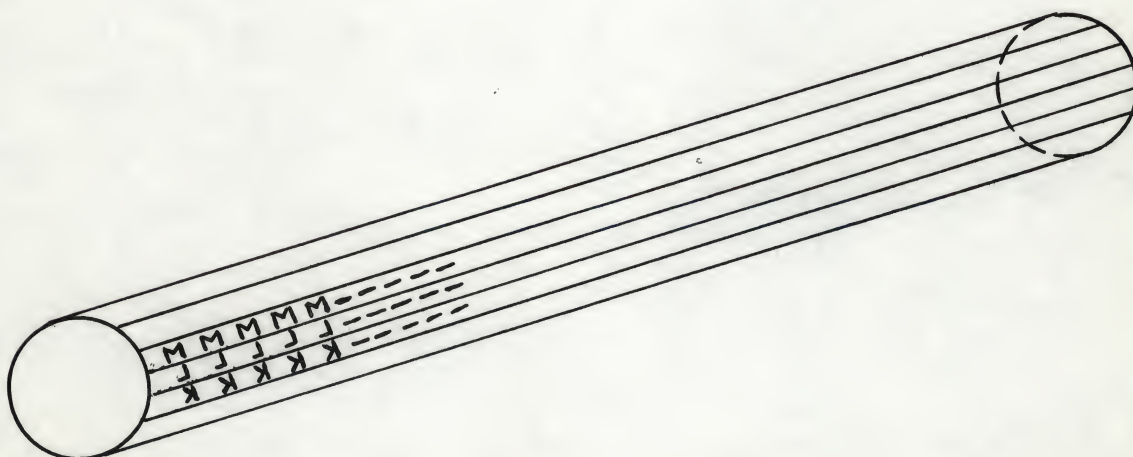
Informatie die geponst moet worden, wordt vanuit het geheugen via een kanaal naar het controle-apparaat van de ponsbandponser gebracht die op zijn beurt ponsmesjes activeert door middel van elektro-magneetjes.

#### 2.5. DE SNELDRUKKER

De sneldrukker is op het ogenblik het meest gebruikte apparaat voor gedrukte uitvoer. Hij drukt 1 complete regel van 120 of 160 symbolen tegelijk. De snelheid varieert van 150 tot 1500 regels per minuut.

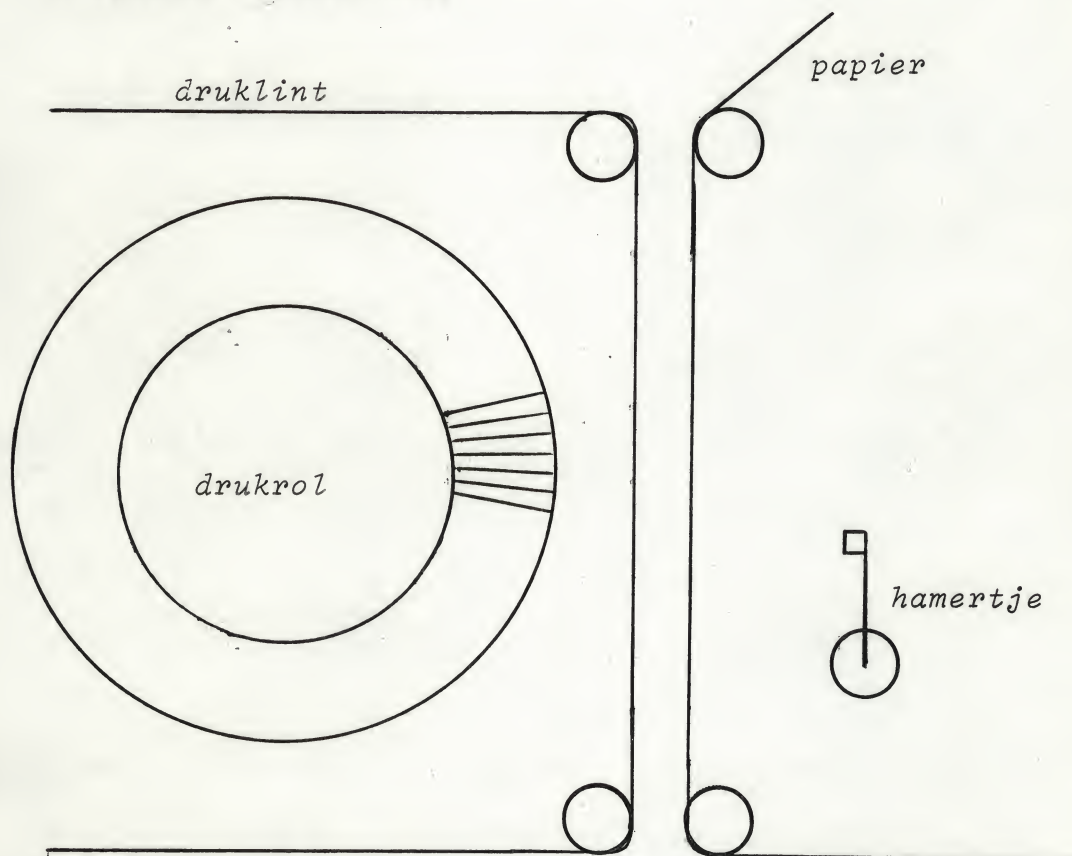
#### WERKING

Er zijn verschillende technische uitvoeringen van sneldrukkers. Een veel toegepaste zal hier worden besproken.





Op een cylinderrol zijn in lengte-richting rijen van 120 karakters aangebracht. Elke rij bevat dezelfde karakters. Het aantal rijen op de omtrek van de rol is afhankelijk van het aantal gekozen symbolen die gedrukt kunnen worden. Hier zal dat niet meer dan 64 bedragen. De rol draait met constante snelheid rond. Vóór de rol bevinden zich 120 hamertjes. Tussen de rol en hamertjes is het papier en druklint aangebracht. (zie bijgaande afbeelding)



Een speciaal register behorende bij de drukker, dat 120 hexaden kan bevatten, wordt vanuit de centrale eenheid gevuld met hexaden, en wel in die volgorde, zoals de tekst straks op het papier moet worden gedrukt. Aan elke hexade in dit register is een hamertje toegevoegd. Op een bepaald ogenblik ligt de rij met bijvoorbeeld de letter A voor. In het register wordt nu gekeken op welke plaatsen de letter A voorkomt en de bijbehorende hamertjes worden tegen het papier gedrukt.

Nu brengt de rol de rij met de letters B voor. De hamertjes die behoren bij de hexade B in het register worden geactiveerd. Enz. Op deze wijze zullen na een volledige omwenteling van de rol alle symbolen die op de rol aanwezig zijn, gepasseerd zijn. Het papier en druklint wordt een regel opgeschoven, het register opnieuw gevuld en het proces herhaalt zich.

#### TESTVRAGEN

1. *Kunt U in het kort de werking van een ponskaartenlezer uitleggen?*
2. *Een kaartponser is langzamer dan een kaartlezer hoe komt dat?*

### 2.6. DE SCHRIJFMACHINE

Veel rekenmachines zijn uitgerust met een schrijfmachine. Deze wordt hoofdzakelijk gebruikt voor de bediening en voor het geven van instructies aan de operateur, bijvoorbeeld voor het verwisselen van magneetbanden, het inleggen van ponsbanden of een pak met ponskaarten.

In het desbetreffende programma wordt de schrijfmachine dan programmatisch opgeroepen.

Meestal bestaat er de mogelijkheid via de schrijfmachine stukken van het geheugen uit te typen. Dit moet echter zoveel mogelijk worden beperkt, gezien de lage uitvoersnelheid. Verder kan via de schrijfmachine informatie in het geheugen worden veranderd. Dit is in sommige gevallen een noodzakelijke, maar zeer gevaarlijke, faciliteit, omdat het kan leiden tot "spelen" met de rekenmachine.

### 2.7. MAGNEETBANDEN

De magneetband is te vergelijken met de band van een bandrecorder. Informatie kan op een band worden geschreven en er later weer vanaf worden gelezen, zoals bij de bandrecorder muziek opgenomen en later weer beluisterd kan worden. De informatie is vastgelegd in binaire vorm.

De magneetband is tot nu toe de meest gebruikte en snelste vorm van in- en uitvoer.

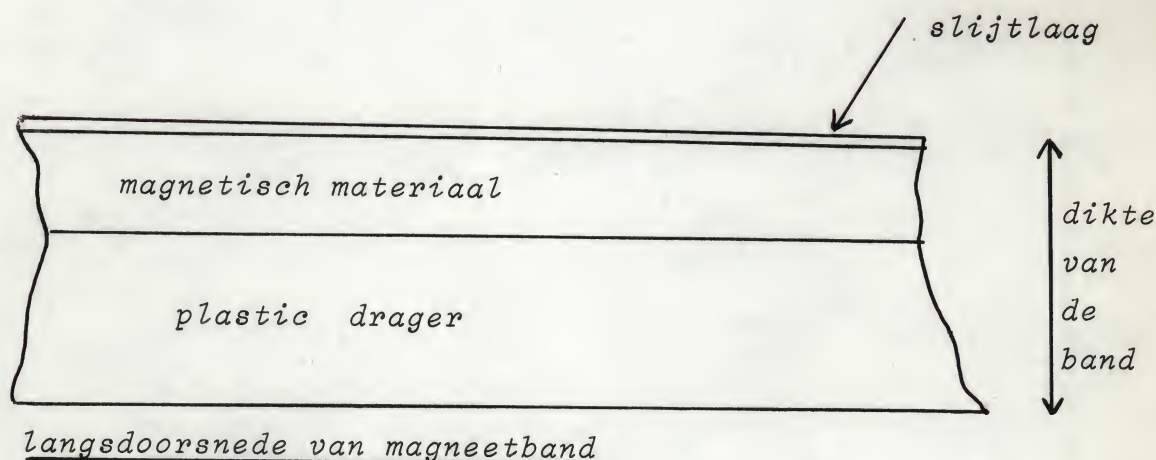
#### PRINCIPE VAN DE MAGNEETBAND

Een magneetband is opgebouwd uit een onderlaag van plastic, waarop een laag magnetiseerbaar materiaal. Vaak is er nog een slijtlaag aanwezig.

#### Enige globale gegevens over een magneetband

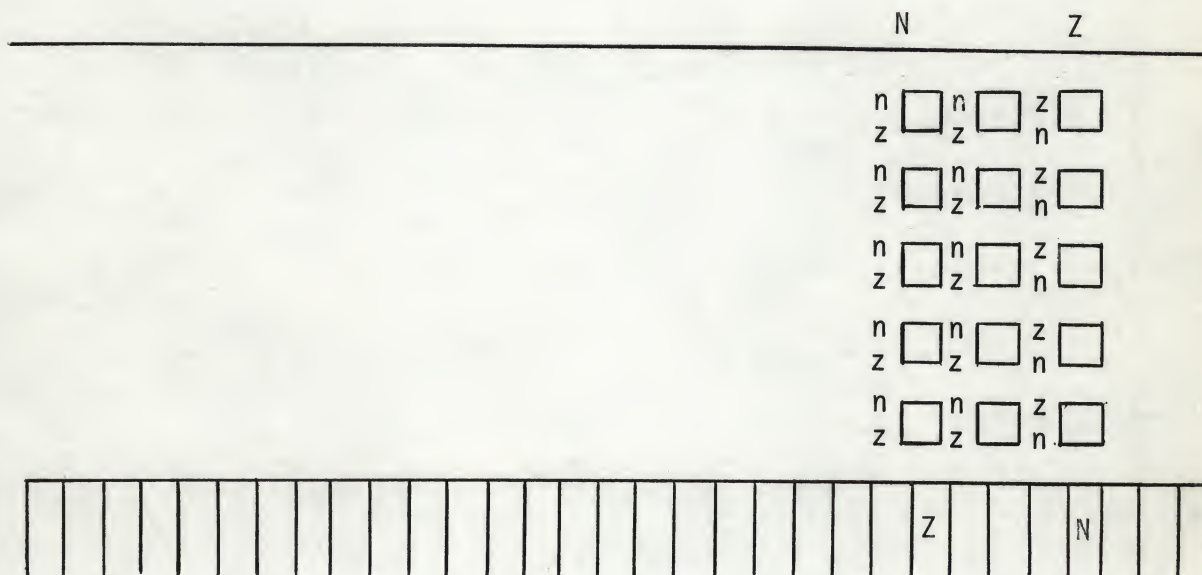
Lengte	750 - 1500 meter
Breedte	$\frac{1}{2}$ , $\frac{3}{4}$ , 1, 2 en 4 inch (1 inch = 2,54 cm)
Dikte	0,2 mm



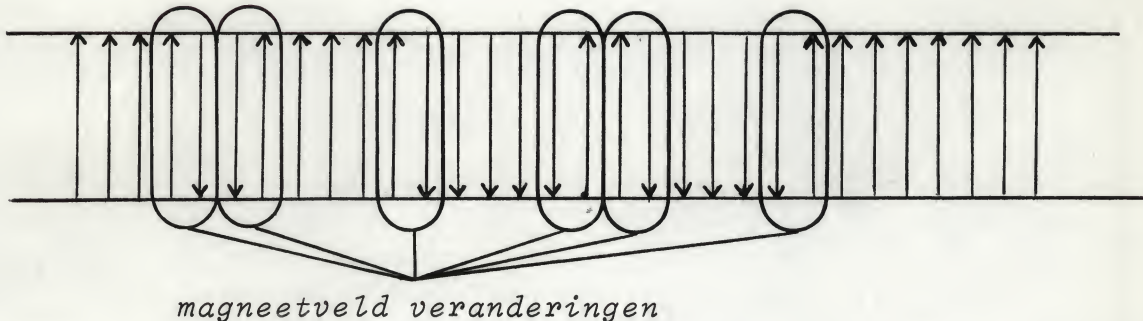


De laag van magnetisch materiaal kan gedacht worden te zijn opgebouwd uit zeer kleine magneetjes. Door middel van een schrijfkop is het mogelijk de stand van deze magneetjes te beïnvloeden. We zullen voor het gemak aannemen dat het magnetisch veld in de schrijfkop dusdanig gericht is, dat de magneetjes in de magnetische laag of met de noordpolen of met de zuidpolen naar boven liggen. Dit is in de praktijk zeker niet waar, maar doet aan het principe weinig af.

een sterk vergrote magnetische laag  
(langsdoorsnede)



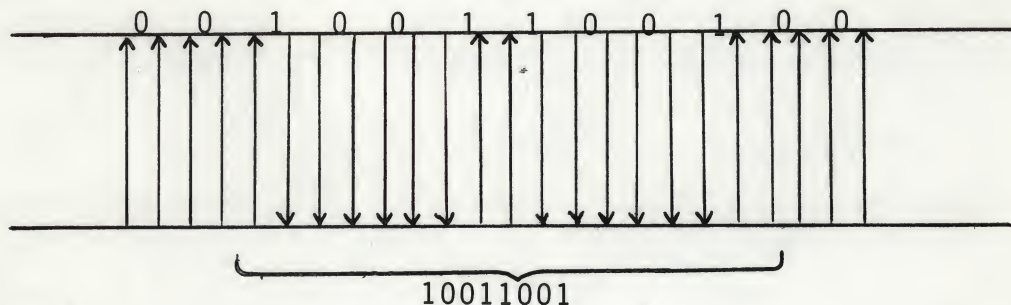
Indien een magneetband beschreven wordt zal de schrijfkop altijd een magnetisch veld opwekken dat in één van de genoemde richting-en werkzaam is. Daarom zullen bij een eenmaal geschreven band alle magneetjes óf met de noordpool naar boven, óf met de zuid-pool naar boven liggen.



Een wisseling van magneetveld richting in de band betekent op die plaats een bit is gelijk aan 1, overal elders zijn de bits gelijk 0.

*N.B. De richting van het magneetveld speelt hier dus geen rol ; slechts de verandering van richting van het veld is belangrijk.*

Als twee pijltjes overeenkomen met een bit op de band dan zal het binaire getal 100011001 schematisch als volgt op de magneetband komen te staan.



#### HET LEZEN VANAF MAGNEETBAND

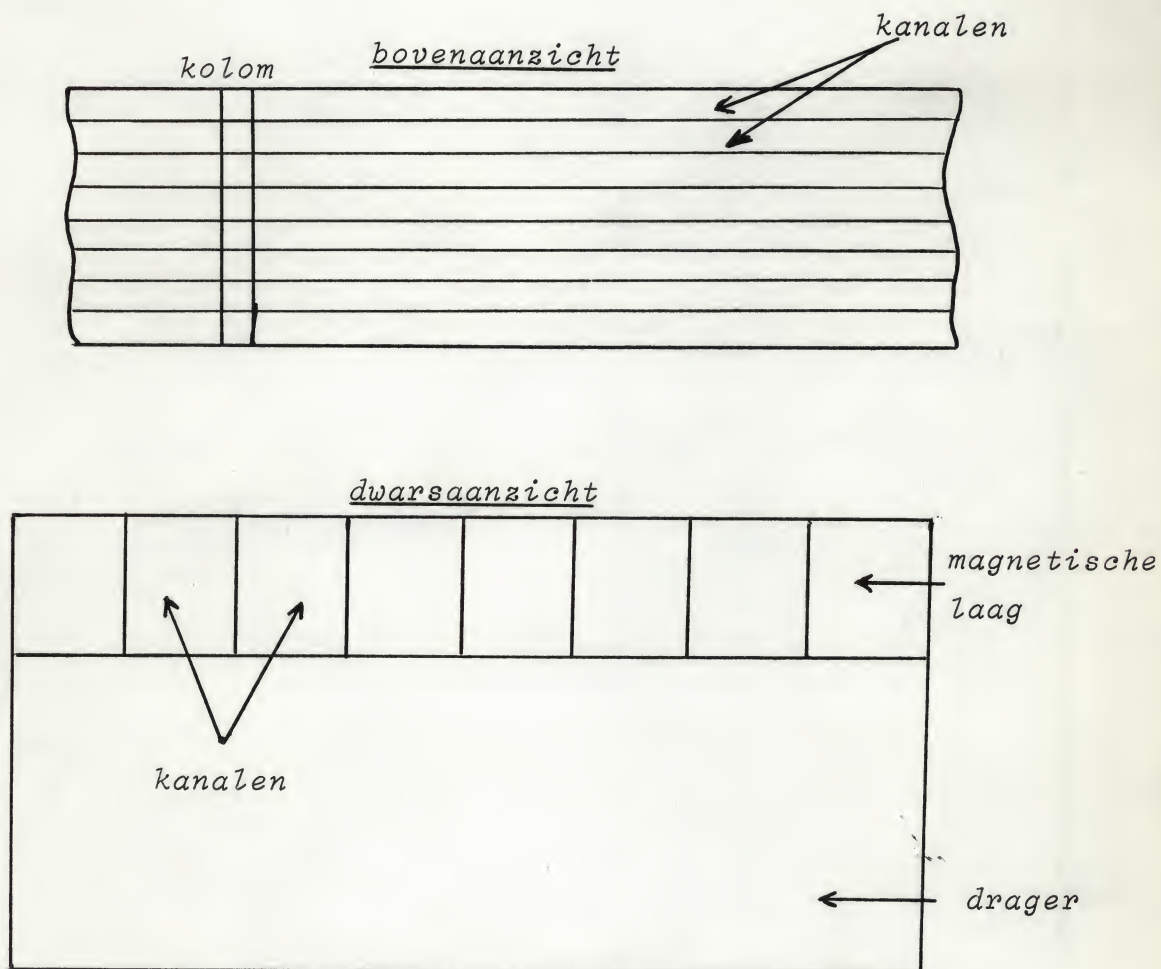
Indien nu de magneetband wordt gelezen, dan zal de leeskop een magnetische veldverandering constateren en dit opvatten als een bit = 1.

Dit schrijven en lezen op een magneetband berust op de inductie theorie uit de elektriciteitsleer, waarop hier niet nader zal worden ingegaan.

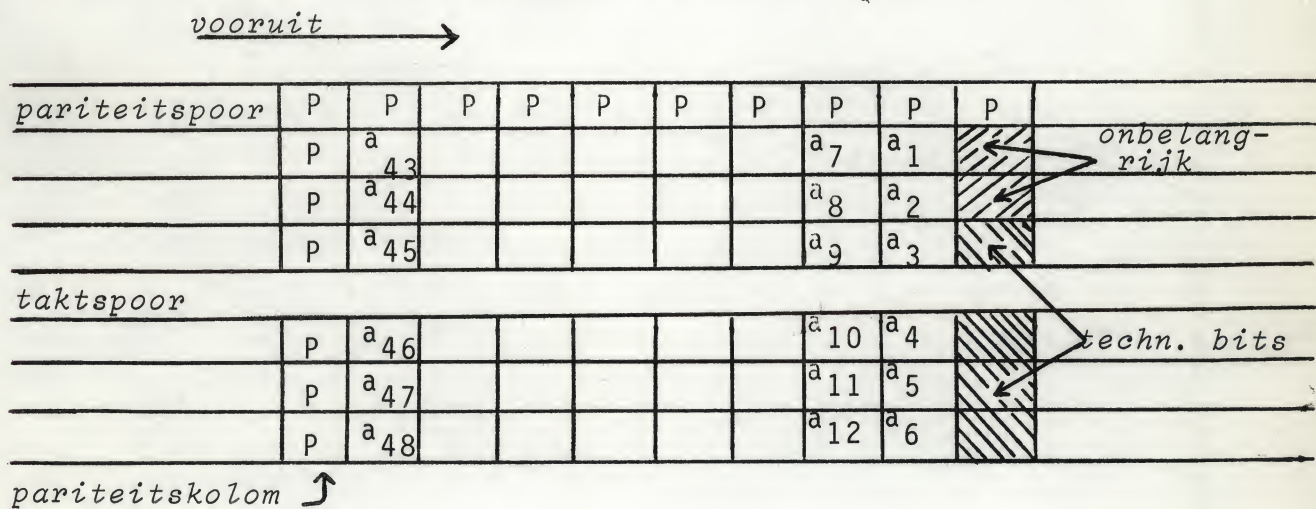
#### TRACKS

Door de breedte van de magneetband is het mogelijk tegelijkertijd meer informatie te schrijven of te lezen. De band is opgebouwd uit een aantal kanalen of sporen (*tracks*), waarop nu tegelijkertijd geschreven of waarvan gelezen wordt volgens het reeds besproken principe.





In de SERA machine wordt gebruik gemaakt van een 8-kanaals-magneetband. Er zal eerst worden bekeken hoe een SERA woord uit het geheugen op de magneetband komt te staan.



## TAKTSP00R

Het vierde spoor van onderen is het taktspoor. Het is te vergelijken met het geleidingsspoor van de ponsband. Het taktspoor geeft telkens aan wanneer er weer een kolom met informatie onder de lezer komt.

De andere sporen, uitgezonderd het bovenste spoor, bevatten programmatische informatie, dus bits van een SERA-woord. Het bovenste spoor is een *pariteitspoor*.

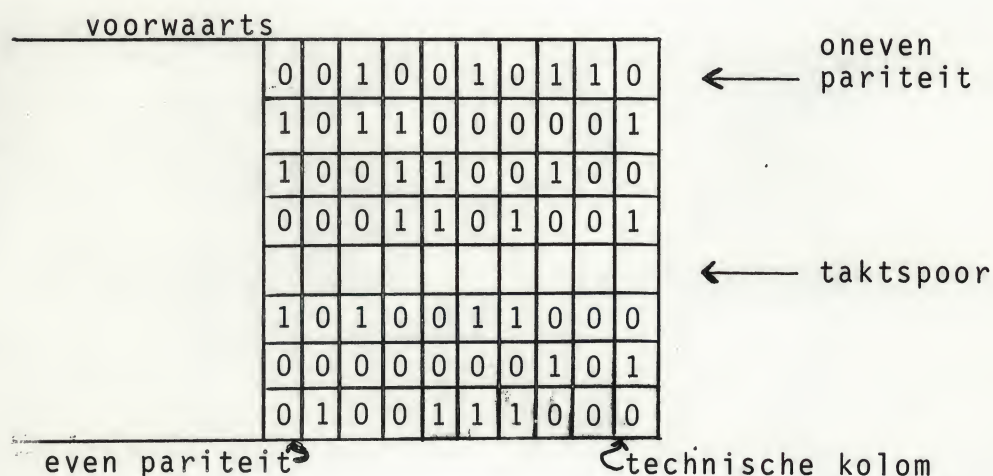
## EVEN EN ONEVEN PARITEIT

Een pariteitbit is een bit die de juistheid van lezen en schrijven test. Dit gaat als volgt te werk. Voordat een woord uit het geheugen op de magneetband gebracht kan worden, moet het in een register worden opgeslagen. In dit register worden de 48 bits verdeeld in groepen van 6 bits die straks tegelijkertijd in een kolom worden geschreven en aan elke groep van 6 wordt een zevende bit, een pariteitbit, toegevoegd. Dit bit wordt 0 of 1, zodanig dat het totaal aantal bits gelijk aan 1 in een kolom oneven wordt. Dit wordt oneven pariteit genoemd. Verder worden alle bits van het woord bekeken die straks op een rij op de magneet komen te liggen. Hieraan wordt een bit toegevoegd zodat de pariteit van zo een rij even wordt. De meest rechtse kolom wordt voor technische doeleinden gebruikt.

Voorbeeld. In een geheugenplaats staat het getal +12345678901.

Dit getal wordt overgebracht naar de magneetband.

(n)	0000	0001	0010	0011	0100	0101	0110	0111	1000
	+	1	2	3	4	5	6	7	8
	1001	0000	0001						
	9	0	1						



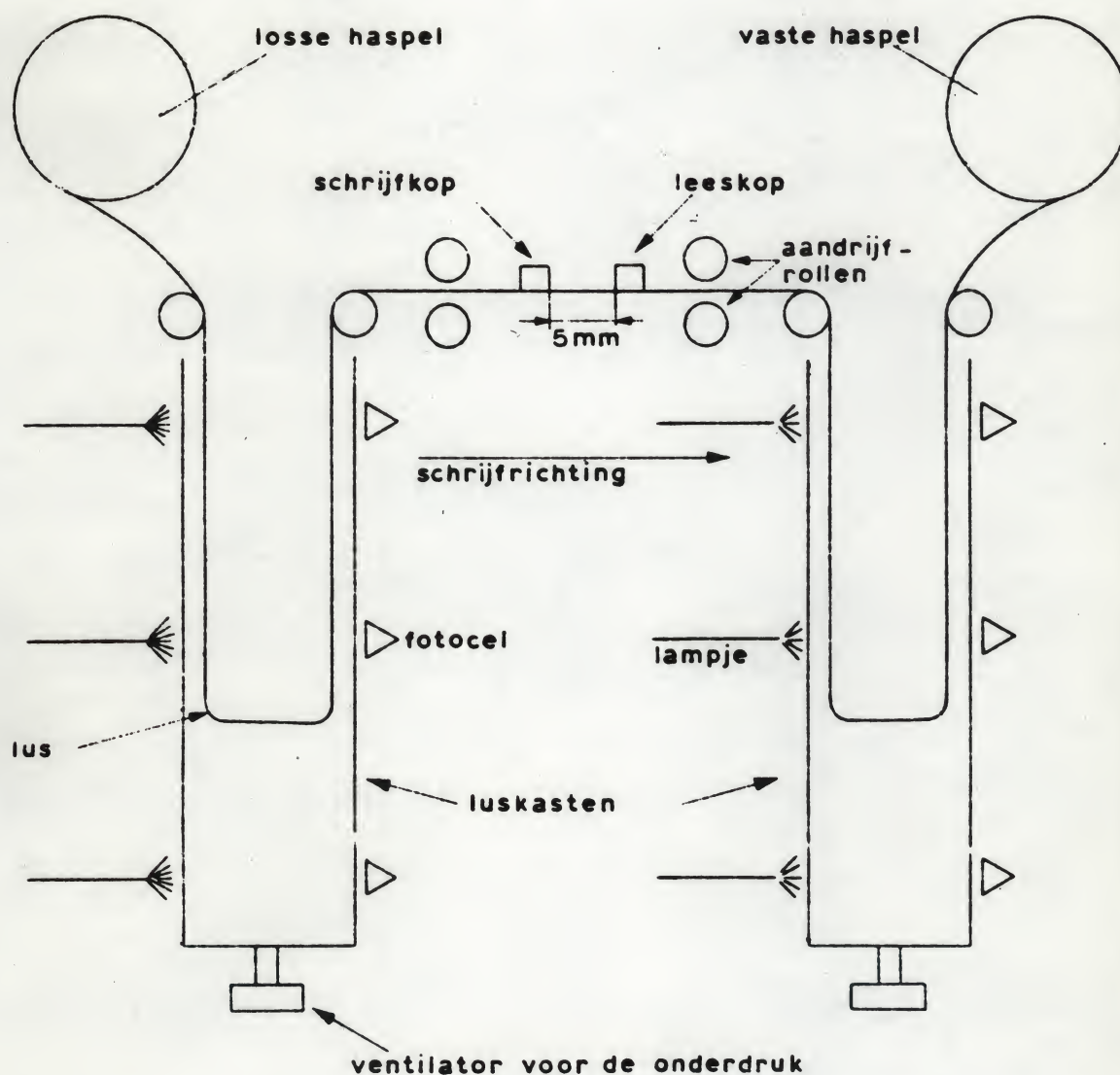
even pariteit

technische kolom



De magneetband is in staat bijzonder veel informatie op te nemen. De hoeveelheid varieert afhankelijk van het fabrikaat en het gebruikte magneetbandapparaat. Hier zal aangenomen worden dat het aantal kolommen 22 per mm bedraagt. Aangezien 10 kolommen een SERA-woord kunnen bevatten, gaan er 22 woorden op een cm. De snelheid van de band is weer afhankelijk van het fabrikaat. Voor de SERA wordt aangenomen : 250 cm. per seconde. Dit betekent een lees en schrijfsnelheid van  $250 \times 22 = 5500$  woorden per seconde.

## 2.8. HET MAGNEETBANDAPPARAAT



Het apparaat werkt met twee haspels, waarvan er één los genomen kan worden van de as (losse haspels). Deze losse



haspel bevat de magneetband waarvan af informatie gelezen of waarop informatie geschreven moet worden.

De magneetband wordt, vanaf de losse haspel, geleid door een aandrijf-mechanisme langs de schrijf- en leeskop en bevestigd aan de vaste haspel. De haspels worden elk door een elektromotor via koppelingen aangedreven. De koppeling zorgt ervoor dat :

- a. de haspel aangedreven wordt,*
- b. de haspel vrijloopt,*
- c. de haspel afgeremd wordt.*

Twee kasten met een rechthoekige doorsnede, waarvan de breedte iets groter is dan die van de magneetband, zorgen ervoor, dat de band door het starten en stoppen niet breekt. In deze kasten hangt n.l. een lus van de magneetband die door een onderdruk naar beneden wordt gezogen. Dit laatste is nodig, opdat de lus een goede vorm krijgt. Het starten en stoppen van de band moet in zeer korte tijd (2 m. sek.) gebeuren. Zonder gebruik van de lussen zouden bij het vertragen en versnellen van de band grote krachten optreden tengevolge van de massa van de haspels, waardoor de band zou breken. In elke kast is in het midden een lampje met fotocel aangebracht om te constateren, of er genoeg band in de kast aanwezig is en de lus dus in stand wordt gehouden. De fotocel stuurt de koppeling van de bijbehorende haspel.

Aangenomen wordt bijv., dat de haspels linksom draaien en dat de lus in de linker kast zo groot is, dat de lichtstraal onderbroken wordt. De linker haspel wordt nu niet aangedreven, maar loopt vrij op zijn massa (freewheelen) Op het moment dat de lus de lichtstraal niet meer onderbreekt wordt de koppeling weer ingeschakeld en de haspel wordt aangedreven. Om deze lussenregeling soepel te laten verlopen zijn, behalve deze fotocellen nog frequentiemeters ingebouwd die de hoeveelheid band/sek. meten die de kast in- en de kast uitgaat en deze hoeveelheden vergelijken. De combinatie van deze meters en de fotocellen regelen het schakelen van de koppelingen. De magneetband moet met constante snelheid langs de lees- en schrijfkop getrokken worden. Hiervoor dienen voor elke richting een paar aandrijfrollen.

De afstand tussen schrijf- en leeskop bedraagt 5 mm. Onder- en bovenaan de kast bevinden zich nog zekeringen die het apparaat doen stoppen indien de lus ondanks de regeling toch nog te lang of te kort wordt.

#### PARITEITSCONTROLE

##### Schrijven

Bij het schrijven op de magneetband worden pariteitsbits



toegevoegd. De leeskop controleert nu direkt na het schrijven of de pariteit in een kolom klopt. Is dit niet het geval dan moet de machine ingrijpen.

#### Lezen

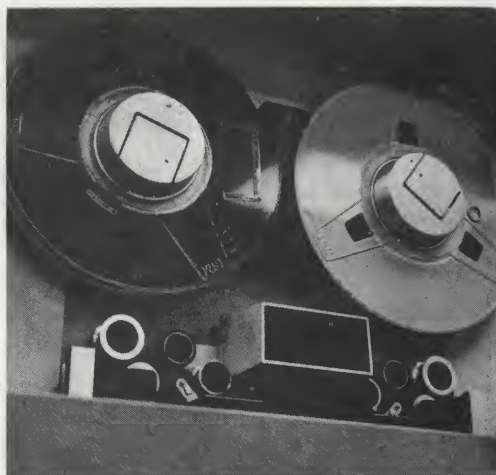
Bij Lezen wordt van een woord zowel de pariteit in een kolom als die in een kanaal getest. Wordt een bit fout gelezen, dan kan door middel van de combinatie pariteitstest in kanaal-richting en in kolomrichting de fout door de machine worden gecorrigeerd. Bij verlies van meer dan één bit is dat onmogelijk.

#### Blokhiaten

Het is voor de magneetband onmogelijk te stoppen precies tussen twee kolommen op de plaats waar men dat wenst. Daarom is het noodzakelijk informatie in blokken bij elkaar te voegen die gescheiden zijn door blokhiaten. Zo'n blokhaat is een stuk op de magneetband, met een lengte van 20 mm, waarop geen informatie staat hetgeen door de leeskop herkend wordt. Deze lengte is o.a. gebaseerd op de tijd nodig om de magneetband op zijn volle snelheid te brengen en om hem af te remmen en te stoppen. De magneetband wordt beschreven vanuit het geheugen. De bijbehorende magneetband-opdrachtgeeft aan hoeveel woorden in een blok vanuit het geheugen geschreven moeten worden. De bloklengte is dus afhankelijk van deze opdracht. Het meest voordelige is natuurlijk een grote bloklengte, dan zijn er minder blokhiaten en dus :

1. de magneetband wordt doelmatig gebruikt
2. het tijdverlies t.g.v. het spoelen over de blokhiaten wordt verminderd.

Een nadeel van een te grote bloklengte is, dat bij fout lezen of schrijven te veel gespoeld moet worden. Dit wordt nog besproken. In vele gevallen wordt de bloklengte bepaald door de aard van het te programmeren probleem.





### 3. PROGRAMMERING VAN DE IN- EN UITVOER

#### 3.1. DE BUFFER, IN VERBAND MET KAARTEN

In het hiernavolgende zal besproken worden welke opdrachten noodzakelijk zijn om informatie in het geheugen te brengen via een van de invoerapparaten en de resultaten uit te voeren op de regeldrukker ponsband of ponskaarten.

Informatie op kaarten moet door een kaartenlezer in het geheugen worden gebracht. Op een of andere manier moet de lezer een opdracht ontvangen die zegt : " *Ga nu een kaart lezen* ". Er zal in het algemeen een stapel kaarten in de lezer klaar liggen waarvan er achtereenvolgens een gelezen moet worden. Zo'n opdracht kan worden gegeven via het bedieningspaneel of door een programma-instructie. Tijdens de afloop van een programma zal het besturingsorgaan, zodra het de leesopdracht ontmoet, de ponskaartenlezer activeren. Er wordt een kaart gelezen, de informatie wordt omgezet (*geconverteerd*) in interne machinetaal en in een daar toe bestemd stuk geheugen gebracht.

In de SERA machine heeft dit stuk geheugen de speciale naam "BUFFER" gekregen.

Een kaart heeft 80 kolommen, waarbij elke kolom een karakter weergeeft. Een kolom wordt nu geconverteerd in een alfanumerieke hexade. De SERA-hexade bestaat uit 6 bits, waarbij elke combinatie van nullen en enen de betekenis heeft van een speciaal karakter (letters, cijfers of speciale tekens). Elke kolom wordt geconverteerd in de daarmee overeenstemmende hexade uit de SERA-hexadencode.

De buffer heeft een zeer speciale vorm, want hij is opgebouwd uit 160 posities, waarbij elke positie een lengte heeft van 6 bits. Deze posities zijn genummerd van 1 t/m 160. Elke kaartkolom wordt, geconverteerd in SERA-hexaden, naar de overeenkomstig genummerde positie gebracht. Kaartkolom 12 bijvoorbeeld komt in positie 12.

De buffer wordt ook gebruikt om informatie vanuit de machine in een kaart te ponsen. De ponsopdracht haalt de alfanumerieke informatie uit de 80 eerste posities van de buffer, converteert in kaartcode en ponsst deze in de 80 kolommen van de kaart. De gehele kaart wordt tegelijkertijd naar de buffer gebracht. bij het lezen, de gehele inhoud van de buffer wordt tegelijkertijd in één kaart gebracht bij het ponsen.

Buffer is een speciale vaste naam in de SERA-code en heeft niets te maken met buffergeheugens die in het vorige hoofdstuk zijn genoemd. Het gebruik van de posities 81 t/m 160 komt later ter sprake.



3.2. DE PONSKAARTOPDRACHTEN

## LSK Lees een kaart

Deze opdracht leest één kaart en brengt de inhoud daarvan over naar de eerste 80 posities van de buffer. Elke positie bevat nu de informatie van één kaartkolom. Deze informatie staat er in de alfanumerieke vorm. De code zoals die op de kaart is geponst wordt dus geconverteerd in SERA-hexaden.

## PSK Pons een kaart

De inhoud van de eerste 80 posities van de buffer, in kaartcode omgezet, wordt geponst in een kaart. Elk karakter uit een van de 80 bufferposities wordt geponst in de daarmee overeenstemmende kaartkolom. De informatie moet in alfanumerieke vorm in de buffer aanwezig zijn.

Het is niet noodzakelijk elke kolom van de kaart te benutten. Een kolom zonder ponsingen wordt bij het lezen geconverteerd in een spatie-hexade.

Voorbeeld. In een kaart staat de volgende informatie geponst:

kolom	de symbolen
3 t/m 7	→ AJ203
8 t/m 11	→ 0183
14 t/m 25	→ J. KLEERMAKER
32 t/m 33	→ 12
50 t/m 55	→ 120.25

In de niet genoemde kolommen is niet geponst, dat zijn dus blanke kolommen op de kaart. De daarbij behorende hexade wordt omgezet in een spatie-hexade en wordt hier voorgesteld door §. Na een LSK opdracht, waarbij de desbetreffende kaart gelezen zal worden, ziet de buffer er uit als in de volgende figuur.

pos	hex	pos	hex	pos	hex	pos	hex	pos	hex	pos	hex	pos	hex	pos	hex
1	§	2	§	3	A	4	J	5	2	6	0	7	3	8	0
9	1	10	8	11	3	12	§	13	§	14	J	15	.	16	R
17	L	18	E	19	E	20	R	21	M	22	A	23	K	24	E
25	R	26	§	27	§	28	§	29	§	30	§	31	§	32	1
33	2	34	§	35	§	36	§	37	§	38	§	39	§	40	§
41	§	42	§	43	§	44	§	45	§	46	§	47	§	48	§
49	§	50	1	51	2	52	0	53	.	54	2	55	5	56	§
57	§	58	§	59	§	60	§	61	§	62	§	63	§	64	§
65	§	66	§	67	§	68	§	69	§	70	§	71	§	72	§
73	§	74	§	75	§	76	§	77	§	78	§	79	§	80	§



### 3.3. SNELDRUKKEROPDRACHTEN

DRU    druk een regel

Deze opdracht heeft geen adres. Op de regeldrukker wordt één regel afgedrukt met symbolen, zoals deze klaar staan in de buffer.

Het aantal symbolen kan 160 bedragen, vandaar een buffer met 160 posities. In deze cursus zal echter aangenomen worden, dat een regeldrukker met 120 tekens per regel is aangesloten.

Dus de opdracht DRU heeft betrekking op de posities 1 t/m 120 van de buffer, die als symbolen van links naar rechts worden afgedrukt. Wil men een aantal nieuwe regels geven, d.w.z. alleen het papier opschuiven, dan kan dit snel gebeuren en wel door de opdracht

DRN    n   druk n nieuwe regels

Het adresgedeelte heeft hier geen adres maar geeft direkt het gewenste aantal nieuwe regels.

### 3.4. PONSBANDOPDRACHTEN

Allereerst 2 opdrachten die betrekking hebben op de buffer.

LSP    n    Lees van ponsband n karakters naar de posities  
          I t/m n van de buffer

PSP    n    Pons ponsband n karakters vanuit de posities  
          I t/m n van de buffer.

Bij deze opdrachten worden letters/cijferwisselingen alsmede terugloopwagen automatisch verwerkt. De samenhang tussen symbolen op ponsbanden en hexaden in de buffer is te vinden op het Overzicht SERA-hexadecode (een bijlage bij deze cursus). In principe kan men deze opdrachten op dezelfde manier gebruiken als de overeenkomstige kaartopdrachten LSK en PSK. Toch moet men met ponsbanden iets anders werken.

Bij een stapel ponskaarten vormt iedere ponskaart een afgesloten hoeveelheid informatie. Iedere ponskaart bevat de nodige gegevens volgens tevoren vastgelegde condities en is bovendien onafhankelijk van de informatie op de volgende kaart.

Bij ponsband daarentegen komt alle informatie direkt achter elkaar en wordt veelal het einde van een informatie-eenheid bepaald door het begin van de volgende. Bij invoeren van gegevens moeten zowel ponskaarten als ponsband eerst met de hand worden geponst. Daarbij is het aanhouden van





*kaartponseenheid*



*centrale verwerkingseenheid met gekoppelde schrijfmachine*



een bepaald formaat , zoals het ponsen van bepaalde cijfers in bepaalde kolommen bij ponskaarten veel eenvoudiger dan bij ponsband, die deze kolom-indeling in het geheel niet kent.

Bij ponsband-invoer maakt men in de regel dan ook gebruik van bepaalde standaardinvoerprogramma's, die de noodzakelijke organisatorische functies verrichten, in plaats van de directe ponsbandopdrachten.

Er zijn enkele gevallen, waarin ponsband de voorkeur verdient boven ponskaarten. Deze doen zich voor indien de gegevens - zonder menselijke tussenkomst - direct geponst worden door de bron. Dit is soms het geval bij technische metingen, waarbij het meetinstrument de gemeten resultaten direct vastlegt in ponsband. In de administratieve sector kan men denken aan ponsapparaten direct gekoppeld aan kasregisters. Vooral bij technische metingen worden de meest verschillende ponscodes gebruikt. Het inlezen in de machine van dit soort banden door middel van de boven omschreven instructies geeft dan moeilijkheden omdat deze opdrachten in de SERA gebaseerd zijn op de telexcode. Bepaalde symbolen, zoals terugloopwagen, letter/cijferwisseling, worden als zodanig niet meer doorgegeven. Verder is deze code gebaseerd op 5-kanaalsband. Juist bij genoemde speciale toepassingen komen vaak 7- en 8-kanaalsband voor.

#### SPECIALE IN- EN UITVOEROPDRACHTEN

Voor dit soort toepassingen bestaan daarom nog enkele speciale in- en uitvoeropdrachten, die we hier vermelden. Bij de volgende opdrachten is p steeds 5,6,7 of 8, afhankelijk van het gebruikte soort ponsband.

INA p Schuif A over p bits naar links. Lees p bits rechts in A

INB p Schuif A en B gekoppeld over p bits naar links. Lees p bits rechts in B

UTA p Pons p bits links uit A. Schuif A en B gekoppeld p bits naar links

UTB p Pons p bits links uit B. Schuif B over p bits naar links.

#### 4. TRANSPORTOPDRACHTEN VAN EN NAAR DE BUFFER

De informatie wordt met behulp van de invoeropdrachten in de buffer gelezen en staat dan in SERA hexaden gereed om verwerkt te worden. Of de informatie is direct in de accu-



mulatoren nodig, of zij moet in het grote geheugen worden opgeslagen. In het laatste geval gaat dit ook via de accumulatoren. Omgekeerd moeten resultaten vanuit het geheugen of direct vanuit de accumulatoren naar de buffer worden gebracht opdat uitvoer mogelijk wordt.

Voor transport van de buffer naar de registers A en B en omgekeerd, zijn de volgende instructies voorhanden.

HAB m:n Haal vanuit de buffer van positie m tot en met n  $\overline{n-m+1}$  hexadenen plaats deze rechts aangesloten in AB. (AB betekent accuA en accuB)

BAB m:n Breng vanuit AB  $\overline{n-m+1}$  hexaden naar de buffer op posities m tot en met n.

De opdrachten HAB en BAB bezitten 2 adressen gescheiden door een dubbele punt. Eigenlijk zijn dit geen adressen, maar duiden m en n posities aan in de buffer. Met deze konstruktie kunnen de gewenste posities zeer gemakkelijk naar accuA en accuB of omgekeerd informatie vanuit de accumulatoren naar de buffer worden gebracht.

Een verdere bijzonderheid is, dat beide accumulatoren bij de opdracht worden betrokken. Dit is vooral van belang indien getallen worden verplaatst van de buffer naar de accu's en omgekeerd, hetgeen bij de behandeling van de converteringsopdrachten duidelijker zal worden.

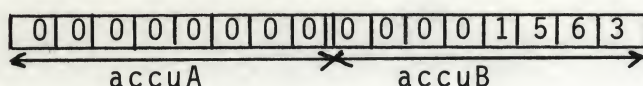
Voorbeeld. In een kaart is in kolom 2 t/m 5 geponst het getal 1563. Gevraagd dit getal in hexaden op te leveren in accuB.

Allereerst moet de kaart ingelezen worden. Dit geschiedt met de opdracht LSK.

De eerste 5 posities van de buffer zien er nu als volgt uit :

positie	hexade
1	8
2	1
3	5
4	6
5	3

Met de opdracht HAB 2:5 worden de hexaden naar accuB gebracht.



Voorbeeld. Gegeven 2 kaarten met de volgende informatie :

<u>kaart 1</u>		<u>kaart 2</u>	
kolom	informatie	kolom	informatie
1 t/m 8	f.pieter	1 t/m 8	j.groots
16 t/m 23	01832616	16 t/m 23	16321801
32 t/m 39	laplands	32 t/m 39	frankrijk

Gevraagd wordt deze 2 kaarten in te lezen en de informatie die erop staat, achter elkaar in het geheugen op te slaan, te beginnen bij adres 1000.

Het programma dat dit verzorgt kan er als volgt uitzien:

100	LSK	lees de 1e kaart
101	HAB 1:8	de eerste 8 posities van de buffer AB
102	BPB 1000	(accu B) → 1000
103	HAB 16:23	positie 16 t/m 23 → AB
104	BPB 1001	(accu B) → 1001
105	HAB 32:39	positie 32 t/m 39 → AB
106	BPB 1002	(accu B) → 1002
107	LSK	lees 2e kaart
108	HAB 1:8	
109	BPB 1003	
110	HAB 16:23	
111	BPB 1004	
112	HAB 32:39	
113	HPB 1005	

Opmerkingen :

1. De opdracht die op adres 107 staat leest de 2de kaart. Hierdoor wordt de informatie van de eerste kaart in de buffer "overschreven"
2. Als het programma voltooid is, zullen de adressen 1000 t/m 1005 er als volgt uitzien :

1000	f.pieter
1001	01832616
1002	laplands
1003	j.groots
1004	16321801
1005	frankrijk

HET SCHOONMAKEN VAN DE BUFFER

Er bestaat de mogelijkheid de buffer "schoon" te maken, d.w.z. een aantal posities te vullen met de hexade die een blank vertegenwoordigt op de ponskaart. De opdracht luidt :

BUS m:n Maak de buffer schoon van positie m t/m n.



De drie opdrachten HAB, BAB en BUS zou men kunnen opvatten als twee adresinstructies werkend op een geheugen van variabele woordlengte van 160 karakters.

De drie opdrachten volgen nu volledig met conventies.

- HAB m:n Haal vanuit de buffer vanaf positie m t/m n in accu A en accu B de inhoud van positie m t/m de inhoud van positie n wordt rechts aangesloten naar AB gebracht. De inhoud van positie n staat in de 6 meest rechtse bits van accu B. De overgebleven ruimte wordt opgevuld met nulhexaden. Indien  $n-m+1 > 16$ , dan worden, teruggeteld vanaf n, 16 hexaden naar AB gebracht. m staat in het normale adresgedeelte van het woord dat de instructie bevat. n staat in het gedeelte van het woord gevormd door de tetraden 2 t/m 5 van links af gerekend. Er moet gelden :  $n \geq m$ ,  $n \leq 160$ ,  $m \leq 160$ . Indien niet aan alle drie bovenstaande voorwaarden is voldaan dan wordt het alarmregister gezet en de opdracht niet uitgevoerd.
- BAB m:n Breng de inhoud van accu A en accu B naar de buffer op posities m t/m n. Indien  $n-m+1 > 16$ , dan worden 16 hexaden naar de buffer gebracht op de 16 posities, teruggeteld vanaf n. Overige posities in de buffer worden niet verstoord. De meest rechtse hexade in B komt op positie n. De inhoud van AB blijft onaangetast. Zie verder HAB m:n
- BUS m:n Maak de buffer schoon vanaf positie m t/m n. De buffer wordt in de aangegeven posities gevuld met spatie-hexaden die op de ponskaart een ongeponste kolom geven en die bij de regeldrukker en ook bij ponsband een spatie vertegenwoordigen.  $n \geq m$ ,  $m \leq 160$ ,  $n \leq 160$ . Indien niet aan een van de bovenstaande voorwaarden is voldaan dan wordt het alarmregister gezet en de opdracht niet uitgevoerd.

## 5. CONVERSIEOPDRACHTEN

### 5.1. INLEIDING

In de buffer staan de gegevens in hexadenvorm. Bestaan deze gegevens uit getallen die worden gebruikt bij rekenopdrachten dan moet er een omwerking (conversie) plaatsvinden van de hexadenvorm van een cijfer of teken naar de tetradenvorm. Een woord in de SERA-code kan op twee manieren geïnterpreteerd worden :

- Alfanumeriek. Dan 8 hexaden in een woord.
- Numeriek. 11 cijfertetraden+1 tekentetrade in een woord.



Een getal, geponst in een kaart, moet in het geheugen worden gebracht. Dan worden achtereenvolgens de volgende bewerkingen uitgevoerd :

1. Lees de kaart in de buffer.
2. Haal vanuit de buffer het getal (in alfanumerieke vorm) en breng het naar de accumulatoren.
3. Converteer het getal van de alfanumerieke vorm naar de numerieke vorm.
4. Breng het getal naar het gewenste adres in het geheugen.

De moeilijke schakel is hier 3. Informatie waarop rekenbewerkingen worden uitgevoerd, moeten intern in de tetradenvorm staan. Alle informatie in de buffer heeft de hexadenvorm. Getallen waarmee gerekend wordt, moeten dus altijd geconverteerd worden na invoer of vóór uitvoer.

#### 5.2. DE OPDRACHTEN

KAG n Konverteer een getal van de alfanumerieke code naar de numerieke code . n bepaalt het aantal te converteren hexaden.

KGA tsn Konverteer een getal in de numerieke code in accuB naar alfanumerieke code in AB. n bepaalt het aantal te converteren tetraden. De letters t en s worden later behandeld.

Een getal neemt in hexadenvorm meer plaats in dan in tetradenvorm. De maximale lengte van getallen waarmee in de machine wordt gerekend, is 11 decimalen, want een woord van 48 bits kan 11 getaltetraden en een tekentetrade bevatten. Wordt zo'n getal vanaf een ponskaart in de buffer gelezen, dan zijn daarvoor 12 posities nodig. 12 posities betekent echter  $12 \times 6 = 72$  bits. Deze 12 posities worden nu naar beide accumulatoren gebracht, waarbij eerst accuB geheel wordt gevuld. De conversie-opdracht verkort de maximaal 12 hexaden tot een getal van maximaal 11 decimalen+teken, waarbij dit getal dan na de conversie geheel in accuB staat.

Voorbeeld. In de buffer staat op de posities 10 t/m 13 het getal 1234. Dit getal moet naar accuB worden gebracht en daar in de SERA getalvorm worden geconverteerd. Het programma zal luiden.

HAB 10:13 Haal getal uit de buffer en breng dit naar accuB

KAG 4 converteer de 4 hexaden.

De 4 posities van de buffer hebben de volgende inhoud :



positie	6 bits
10	000001
11	000010
12	000011
13	000100

De HAB opdracht vult de accumulatoren, waarbij hier alleen de inhoud van accuB wordt getoond. De inhoud van accuA zal uit nul bits bestaan.

accuB

000000	000000	000000	000000	000001	000010	000011	000100
--------	--------	--------	--------	--------	--------	--------	--------

Na de KAG opdracht ziet de B-accumulator er als volgt uit :

accuB

0000	0000	0000	0000	0000	0000	0000	0000	0001	0010	0011	0100
------	------	------	------	------	------	------	------	------	------	------	------

De KAG opdracht verkort het getal dus, snijdt van ieder cijfer twee nul bits af.

Voorbeeld. In de buffer staat op de posities 1 t/m 10 het getal +123455897.  
Dit getal moet in de tetradenvorm naar adres 100 gebracht worden.

Het stuk programma die dit zal uitvoeren luidt :

```
HAB 1:10 eerste 10 posities → AB
KAG 10  converteer 10 hexaden
BPB 100 getal → 100
```

De register accuA en accuB voor de converteringsopdracht en accuB na de converteringsopdracht volgen nu.

accuA				accuB							
00000000	111100	000001	000010	000011	000100	000101	000101	000101	001000	001001	000111

accuB											
0000	0000	0000	0001	0010	0011	0100	0101	0101	1000	1001	0111

Voorbeeld. Op positie 10 t/m 14 staan de hexaden : -3462  
Dit getal moet geconverteerd worden. AccuB voor en na de converteringsopdracht volgen hier.



accuB

000000	000000	000000	101100	000011	000100	000110	000010
--------	--------	--------	--------	--------	--------	--------	--------

accuB

1111	1111	1111	1111	1111	1111	1111	0011	1100	1011	1001	1101
------	------	------	------	------	------	------	------	------	------	------	------

Bij de convertering wordt zoals men kan zien, rekening gehouden met het teken. De hexade (-) zorgt er voor dat de converteringsopdracht het getal in de juiste negatieve getalrepresentatie brengt.

#### VOORBEELD VOOR DE HAB- EN KAG-OPDRACHT.

Een groothandel in kruidenierswaren verwerkt de voorraad-administratie en verkoopstatistieken op een elektronische rekenmachine. Daartoe worden een aantal gegevens op een ponskaart bijeengebracht, waarbij bepaalde gegevens aan een rekenproces deelnemen, andere dienen voor vergelijking of bijkomende informatie, zoals bijvoorbeeld naam van de koper of verkoper. Aan de berekening zelf zal geen aandacht worden besteed, slechts de invoer zal worden besproken. Bepaalde informatie zal moeten worden geconverteerd, bij andere zal dat niet noodzakelijk zijn, afhankelijk of ze aan het rekenproces deelnemen of niet. De ponskaart bevat de volgende informatie, waarbij een aantekening is gemaakt, indien er geconverteerd moet worden.

#### kolom

1-16	naam van de verkoper	
19-20	rayonnr. van de verkoper	
23-24	nr. van de verkoper	
27-42	naam van de koper	
45-47	nr. van de koper	
50-54	artikelnr.	
57-60	aantal van het verkochte artikel	/convert.
63-68	prijs per stuk van het artikel in centen	/convert.

#### In dit voorbeeld:

#### kaartinhoud naar geheugenadressen :

naam verkoper	p.vlotjan	→	100	en	101
rayonnr.	11	→	102		
nr.verkoper	08	→	103		
naam koper	j.droogstoppel	→	104	en	105
nr.koper	084	→	106		
art.nr.	01023	→	107		
aantal	0200	→	108		
prijs	000125	→	109		



Ir. P.A. Tas

Het programma ziet er als volgt uit :

10	LSK		lees een kaart
11	HAB	1:16	pos. 1 t/m 16 → AB
12	BPA	100	8 hexaden → 100
14	BPB	101	8 hexaden → 101
15	HAB	19:20	pos. 19 t/m 29 → B
16	BPB	102	rayonnr. → 102
17	HAB	23:24	pos. 23 t/m 24 → B
18	BPB	103	nr. verkoper → 103
19	HAB	27:42	pos. 27 t/m 42 → AB
20	BPA	104	8 hexaden → 104
21	BPB	105	8 hexaden → 105
22	HAB	45:47	pos. 45 t/m 47 → B
23	BPB	106	nr. koper → 106
24	HAB	50:54	pos. 50 t/m 54 → B
25	BPB	107	art. nr. → 107
26	HAB	57:60	pos. 57 t/m 60 → B
27	KAG	4	converteer van hexaden naar tetraden
28	BPB	108	aantal → 108
29	HAB	63:68	pos. 63 t/m 68 → B
30	KAG	6	converteer van hexaden naar tetraden
31	BPB	109	prijs → 109

De opdrachten KAG en KGA werken in de beide accumulatoren A en B. De n in het adresgedeelte heeft geen betrekking op een adres, maar duidt het aantal te converteren cijfers aan met inbegrip van een teken. Een getal in hexadenvorm kan voorafgegaan worden door een tekenhexade.

Bij het converteren naar de tetradenvorm moet het teken juist verwerkt worden. Bevat het aantal te converteren hexaden geen tekenhexade dan wordt aangenomen dat het getal positief is en wordt als zodanig in tetraden voorgesteld. In het omgekeerde geval, dus indien tetraden in hexaden worden omgezet, is er altijd een tekentetrade aanwezig. Het is echter in vele gevallen wenselijk dit teken te onderdrukken, dus dat de tekentetrade niet omgezet wordt in een tekenhexade.

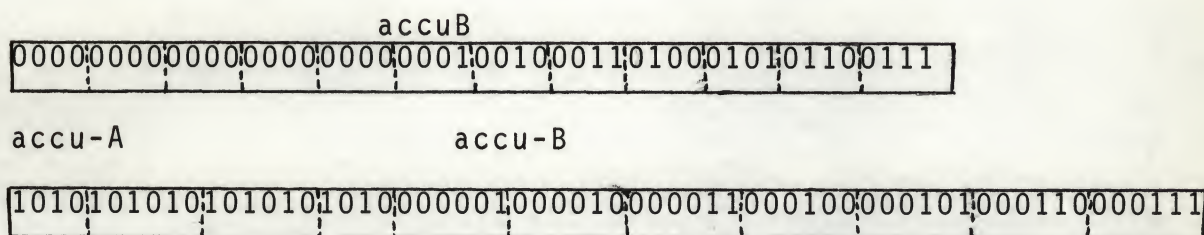
Bij de KGA-opdracht kan daarom in het adresgedeelte de letter t toegevoegd of weggelaten worden, al naar gelang het teken niet of wel onderdrukt moet worden.

Verder kan het heel goed mogelijk zijn dat n behorende bij KGA tsn groter is dan het aantal te converteren significante cijfers ( met significante cijfers wordt bedoeld: cijfers met betekenis +00234=+234. Hier zijn de twee nullen niet significant) Door de letter s toe te voegen worden in plaats van de niet significante nulhexaden, spatiehexaden opgeleverd. Dit is van belang als de uitvoer wordt omgezet in gedrukte

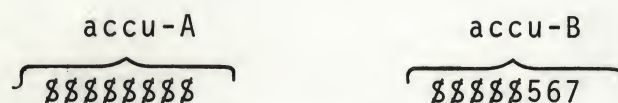


tekst en niet-significante nullen niet gedrukt mogen worden. In plaats van het getal 001234 wordt dan op papier gedrukt het getal 1234. Op de plaats van de nullen zijn dan spaties gekomen.

Voorbeeld. Het getal +00001234567 wordt door de opdracht KGA 7 omgezet in de hexaden 1234567. De rest van de accumulatoren wordt opgevuld met spatie-hexaden.



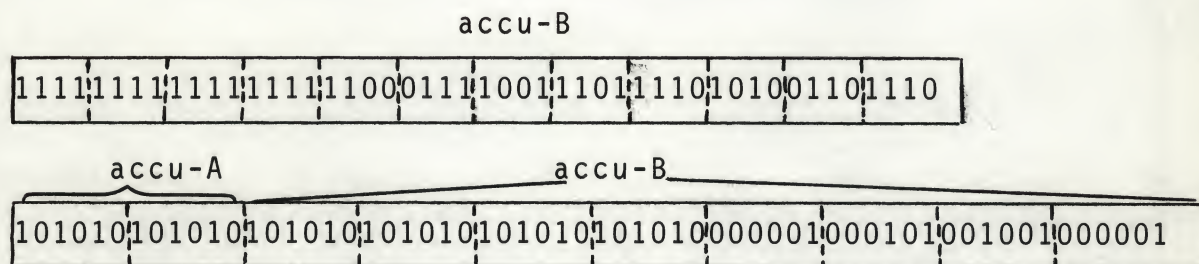
KGA 3 zal het getal in accu-B omzetten in de 16 hexaden in de accumulatoren.



Is het getal in accu-B negatief dan zorgt KGA opdracht er toch voor dat de juiste hexaden opgeleverd worden. Een negatief getal gebruikt een andere cijferrepresentatie als een positief getal.

Als een getal bekeken wordt in de hexadenvoorstelling, dan is er geen verschil tussen positieve of negatieve cijfers. Het positief of negatief zijn van een getal wordt immers aangegeven door de tekenhexade.

Voorbeeld. -00038621591 wordt door de opdracht KGA 4 omgezet in de 4 hexaden 1591 en de rest van de accumulatoren zijn dan spatie-hexaden.



Uit de voorbeelden blijkt dat de negatieve en positieve representatie van een cijfer in dezelfde cijferhexaden worden omgezet. Nu is het begrijpelijk dat men ook wel geïnteresseerd is in het teken van een getal. Bij de KGA opdracht kan





*Gegevens uit vele duizenden ponskaarten kunnen  
op één enkele magneetband worden vastgelegd*



*magneetbandeenheden*



daarom in het adres gedeelte de letter t toegevoegd of weggelaten worden, al naar gelang het teken niet of wel onderdrukt moet worden. De tekenhexaden wordt meegeteld in de letter n.

Voorbeeld. In accu-B staat het getal +00056218369. De opdracht KGA t4 geeft rechts aangesloten in accu-B de hexaden +369. De rest van de accumulatoren is opgevuld met spatie-hexaden.

accu-B

0000	0000	0000	0000	0101	0110	0010	0001	1000	0011	0110	1001
------	------	------	------	------	------	------	------	------	------	------	------

accu-B

0101	0101	0101	0101	0101	0101	1111	0000	1100	1001	1001	1001
------	------	------	------	------	------	------	------	------	------	------	------

Voorbeeld. In B staat het getal 12345 tetradenvorm.

+ 0 0 0 0 0 0 1 2 3 4 5

Door een KGA opdracht wordt dit getal geconverteerd in een aantal hexaden in AB, afhankelijk van tgn. Achter de opdracht zijn de gevormde hexaden geschreven.

KGA ts6	§	§	§	§	§	§	§	+	1	2	3	4	5
KGA s7	§	§	§	§	§	§	+	§	1	2	3	4	5
KGA s6	§	§	§	§	§	§	§	§	1	2	3	4	5
KGA s5	§	§	§	§	§	§	§	§	1	2	3	4	5
KGA t8	§	§	§	§	§	+	0	0	1	2	3	4	5
KGA 6	§	§	§	§	§	§	§	0	1	2	3	4	5
KGA 2	§	§	§	§	§	§	§	§	§	§	§	4	5
KGA t10	§	§	§	+	0	0	0	0	1	2	3	4	5
KGA	§	§	0	0	0	0	0	0	1	2	3	4	5

accu-A
accu-B

*N.B. Indien de letter t aanwezig is, dan wordt de tekenhexade meegeteld voor het totale aantal geconverteerde hexaden. Bij KGA t8 worden 7 cijfers en een teken geconverteerd.*

De conversie-opdrachten volgen nu volledig met conversies.

KGA tsn Konverteer een getal in numerieke code in register B naar alfanumerieke code in AB. n bepaalt het aantal geconverteerde hexaden.  
 letter t afwezig : het teken wordt onderdrukt  
 letter t aanwezig: het teken wordt in een tekenhexade geconverteerd en voor het getal gevoegd.



letter s aanwezig : niet significante nullen worden  
geconverteerd in spatie-hexaden  
letter s afwezig : niet significante nullen worden  
nulhexaden.

1. Indien het te converteren getal in B niet voldoet aan de eisen voor een decimaal getal, dan wordt de opdracht niet uitgevoerd en het alarmregister gezet.
2. De  $(16-n)$  overige hexaden in AB worden gelijk aan spatie-hexaden.
3. Indien t aanwezig dan  $n \leq 12$   
Indien t afwezig dan  $n \leq 11$   
Is hieraan niet voldaan, dan wordt n respectievelijk gelijk genomen aan 12 of 11.
4.  $n=0$ . Het gehele B register wordt geconverteerd, met inachtneming van bovengenoemde regels.

KAG n    Konverteer een getal van alfanumerieke code naar de numerieke code. Het getal staat in AB en n bepaalt het aantal te converteren hexaden, die rechts aaneengesloten in AB staan. Het geconverteerde getal komt rechts aaneengesloten in accu-B. Eventuele hexaden die een spatie aangeven, worden genegeerd, doch wel meegerekend voor n. De conversie wordt van rechts naar links in de accumulator uitgevoerd. De opdracht wordt in de volgende gevallen als voltooid beschouwd:

1. Er zijn n hexaden geconverteerd. Was de  $n^e$  hexade een tekenhexade, dan bepaalt die het teken van het geconverteerde getal. In elk geval wordt het teken positief gemaakt.
2. Tijdens het converteren wordt een tekenhexade ontmoet. Het geconverteerde getal krijgt het desbetreffende teken.
3. Een van de n hexaden is geen cijfer, teken of spatie-hexade. Het geconverteerde getal krijgt het positieve teken.
4. Er zijn reeds 11 getalhexaden ontmoet. Indien  $n=0$  of  $n \geq 16$ , dan wordt de convertering toegepast alsof  $n-16$  is.

## 6. KORTE BESCHOUWING OVER DE IN- EN UITVOER.

De bewerkingssnelheden van de moderne rekenmachine liggen in de orde van microseconden. De snelheid van in- en uitvoer verschilt daarmee zeker een faktor 1000. Men zal dus trachten de in- en uitvoer te versnellen. Toch zal het verschil tussen snelheden in de rekenmachine en de in- en uitvoer-apparatuur wel altijd groot blijven. De laatste immers werkt ten dele mechanisch.



In- en uitvoer baart bij het gebruik van rekenmachines in de administratie nog steeds de grootste zorgen. Dit geldt vooral voor de invoer. In het algemeen zullen vrij grote hoeveelheden gegevens, die geschreven of getypt op papieren documenten vastgelegd zijn, geponst moeten worden op ponskaarten of ponsband.

Ponskaarten verdienen hier meestal de voorkeur, omdat elke kaart een hoeveelheid informatie apart en onafhankelijk van de andere kaarten is waardoor correctie van geponste fouten zeer eenvoudig is. Bovendien kan met behulp van de kolommen een bepaalde lay-out worden toegepast.

Nadeel van ponskaarten is, dat zij aanzienlijk meer ruimte innemen dan ponsband en dat de ponskaarten-apparatuur veel duurder is dan ponsband-apparatuur.

Het is duidelijk, dat het ponsen van informatie een tijdrovende en fout verwekkende procedure is. Daarom wordt intensieve studie gemaakt van andere mogelijkheden van invoer. Er wordt hier gedacht aan :

- a. *het lezen van met magnetische inkt gedrukte symbolen*
- b. *optisch lezen van speciale karakters*
- c. *mark sensing.*

a en b worden bijvoorbeeld toegepast in het cheque-verkeer van banken. Mark sensing is een aanduiding voor het aanstrepen van kaarten met behulp van een grafiet potlood. Deze aangestreepte kaarten kunnen door een speciale lezer gelezen en omgezet worden in ponskaarten. Het wordt bijv. toegepast door gas- en elektriciteitsbedrijven bij het opnemen van de meterstand.

Magneetbanden kunnen zeer snel informatie overdragen. Het zal alleen praktisch nooit voorkomen dat de primaire informatie, komende van de administratieve sektor in het bedrijf, op een magneetband staat, zodat dan toch eerst van kaarten of ponsband gebruik gemaakt zal moeten worden. De magneetband wordt dan ook meer als geheugen gebruikt. Gegevens over b.v. voorraden, personeel, klanten etc. kunnen er opgebracht en op bepaalde tijden gemuteerd worden.

Er gaat door de in- en uitvoer veel kostbare machinetijd verloren. Teneinde dit op te vangen kan in moderne machines parallel gewerkt worden. Dit betekent tegelijkertijd rekenen en in- en uitvoeren. Hierop zal, bij de bespreking van de magneetbanden, nader worden ingegaan.



Deze bijlage dient ter ondersteuning van Uw studie. M.a.w. U hoeft dit niet uit het hoofd te leren. Het is een verzameling van alle in de programmeringskursus gebruikte Sera-instructies, en kan derhalve tijdens Uw studie dienst doen als naslagwerk.

#### DE SERA-CODE

Bij het onderwijs in de rekenmachinetechniek en daarvan speciaal de programmeertechniek is het noodzakelijk, dat men zich bedient van de code van een of andere machine. Neemt men hiervoor een van de bestaande rekenmachines dan wordt men enerzijds gekonfronteerd met de moeilijkheid, dat men bepaalde details van die machine moet bespreken die uit onderwijsoogpunt niet direkt noodzakelijk zijn. Anderzijds ontbreken weer mogelijkheden, die bij andere machines weer wel aanwezig zijn en die men ook wil behandelen. Om deze redenen heeft men in het verleden reeds meermalen zijn toevlucht genomen tot de code van een hypothetische machine, waarin men dan alle eigenschappen aanbracht, die men als docent noodzakelijk achtte om te bespreken. Dit is ook gebeurd t.b.v. verschillende programmeerkursussen, lezingen en voordrachten verzorgd door de Stichting Studiecencentrum voor Administratieve Automatisering. Deze hypothetische machine kreeg aanvankelijk de naam Stichting Eenvoudige Rekenautomaat, terwijl later de naam SERA ook wel gebruikt werd als afkorting van Schijnbaar Eenvoudige Rekenautomaat. De ervaringen en medewerking van verschillende docenten in de loop van een aantal jaren en het streven SERA zo goed mogelijk aan te passen aan de huidige stand van de techniek van rekenmachines hebben via verschillende wijzigingen geleid tot de huidige samenstelling zoals hierna beschreven. Deze samenstelling is thans bovendien van dien aard dat SERA niet meer alleen gebruikt wordt voor kursussen van de Stichting Studiecencentrum voor Administratieve Automatisering maar b.v. ook bij enkele colleges aan de Technische Hogeschool te Delft. Mede door dit ruimere gebruik van SERA is uit een oogpunt van praktische oefening weer de behoefte ontstaan iets van het volkomen hypothetische van deze machine weg te nemen. Dit is gerealiseerd door voor enkele bestaande machines een interpreteerprogramma voor SERA-code samen te stellen. Daarbij is het natuurlijk niet de bedoeling werkelijk uitgebreide berekeningen via SERA-code door deze machines te laten uitvoeren, maar wel kleine proefprogramma's door de machine "echt" te laten testen. Volledigheidshalve dient te worden vermeld, dat de uitvoering van de SERA-code zoals hier beschreven ook enigszins is beïnvloed door praktische overwegingen i.v.m. een dergelijk interpreteerprogramma. Bij name geldt dit ten aanzien van de woordlengte van 48 bits, die hier gebruikt wordt. Dit past natuurlijk gemakkelijk in een machine die zelf ook werkt met een woordlengte van 48 bits. Deze



woordlengte is overigens voor SERA niet essentieel en het is zonder meer mogelijk t.b.v. interpretatie op een machine met andere woordlengte een daaraan aangepaste SERA-code te gebruiken. Verschillende SERA-instructies krijgen dan vanzelfsprekend een aangepaste betekenis. De structuur van de SERA-machine als zodanig verandert daardoor echter niet. Zoals uit het voorgaande reeds valt op te merken is SERA gebaseerd op een machine met vaste woordlengte. Uit didactische overwegingen is de voorkeur gegeven aan programmatisch onderwijs dat begint met het begrip vaste woordlengte en pas in een later stadium overgaat tot behandeling van variabele woordlengte. Een aanknopingspunt tot het begrip variabele woordlengte vindt men overigens in de huidige SERA-uitvoering door het gebruik van de "buffer" bij in- en uitvoer.

## 1. WOORDINDELING, GEHEUGEN- EN REKENORGAAN

1.0 SERA werkt met een vaste woordlengte van 48 bits. Afhankelijk van het gebruik worden deze bits op verschillende manieren samengevoegd.

### 1.1 Decimaal vast komma-getal

De 48 bits zijn verdeeld in 12 tetraden (4 bits). De meest linkse tetraade is het teken. De overige 11 tetraden vormen 11 decimale cijfers. De cijfervoorstelling per tetraade is zuiver binair als volgt:

0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

Is het getal positief dan bestaat de tekentetraade uit 4 maal 0. Is het getal negatief dan worden alle 48 bits van het woord geïnverteerd, dus ook de getaltetraden. De tekentetraade bestaat dan automatisch uit 4 enen. De cijfertetraden zijn in dit geval.

0	1111	5	1010
1	1110	6	1001
2	1101	7	1000
3	1100	8	0111
4	1011	9	0110

Wanneer bij een voorwaardelijke sprongopdracht getest wordt op de tekentetraade en deze is niet 4 maal 0 of 4 maal 1, dan is de meest linkse bit van de tekentetraade maatgevend voor de afloop van de spronginstructie.

### 1.2 Alf numerieke woordindeling

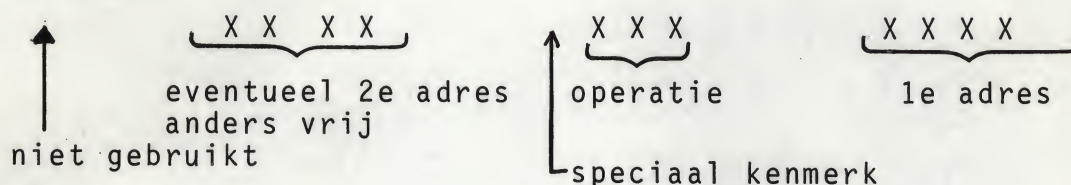
De 48 bits worden nu verdeeld in 8 hexaden (6 bits). Iedere



hexade stelt een bepaald symbool voor, waarvan de betekenis is vermeld in hoofdstuk 12, waarbij tevens het verband met ponskaart en papierbandcode gegeven is.

### 1.3. Instructievorm en geheugenindeling

Het woord geeft eigenlijk weer de getalvorm als 12 tetraden. Deze tetraden worden nu echter in groepen samengevoegd met de volgende betekenis.



SERA werkt in het algemeen als éénadres-machine met max. 10.000 geheugenplaatsen van 48 bits, waarbij het adresgedeelte van een instructie wordt aangegeven op de 4 rechtse tetraden. Het operatiegedeelte van een instructie is een decimaal getal van 2 cijfers. Via het invoerprogramma kan gebruik gemaakt worden van een mnemotechnische code van 3 letters, die één op één afgebeeld wordt in dit operatiegedeelte.

De kenmerktetrade speelt een rol in de volgende gevallen.

Verschillende opdrachten bieden de mogelijkheid om met het le adresgedeelte zelf als operand te werken. Bij het programmeren geeft men dit aan door tussen opdracht en adres een sterretje (\*) te schrijven. In de volgende hoofdstukken is bij de definitie van de opdrachten door een sterretje aangegeven, welke opdrachten van de faciliteit gebruik kunnen maken. SERA werkt zowel met een levend als een dood geheugen. Beide geheugens zijn genummerd van 0000 t/m 9999 en worden aangegeven in het le adresgedeelte. Ook enkele registers in rekenorgaan en besturingsorgaan en andere delen van de machine zijn adresseerbaar. Deze zijn daartoe genummerd en worden in voorkomende gevallen ook in het adresgedeelte aangegeven. De kenmerktetrade geeft bij iedere instructie aan welke toepassing in dat geval nodig is. Deze kan daartoe de volgende samenstelling met bijbehorende betekenis hebben.

0000	gebruik van levend geheugen
0001	le adresgedeelte als operand
0010	gebruik van speciale registers
0100	gebruik van dood geheugen

Gebruikt men het normale invoerprogramma dan wordt deze kenmerktetrade automatisch goed ingevuld. In bijzondere gevallen waarbij programmatisch gemanipuleerd gaat worden met instructies moet hiermee eventueel rekening gehouden worden.



Naast de aangegeven geheugenvormen met woorden van 48 bits is er speciaal t.b.v. in- en uitvoer een zgn. buffergeheugen. Dit bestaat uit 160 hexaden genummerd van 1 t/m 160. Iedere positie van 6 bits in dit buffergeheugen kan dus 1 alfa-numeriek symbool bevatten. Men kan in voorkomende gevallen werken met een groep van opeenvolgende symbolen in dit buffergeheugen, waarbij de lengte van deze groep overigens variabel is. Opdrachten, die betrekking hebben op deze buffer kennen daarom 2 adressen, die het eerste en het laatste adres van de gewenste groep symbolen aangeeft. Het eerste adres staat daarbij op de normale aangegeven 1e adresplaats en het tweede adres op de 4 tetraden links van de kenmerk-tetrade. Bij het gebruik van het normale invoerprogramma kunnen deze twee adressen aangegeven worden door 2 getallen gescheiden door een dubbel punt. B.v. H A B 30:37

Deze opdracht heeft dan betrekking op de adresplaatsen 30 t/m 37 in het buffergeheugen. Het buffergeheugen kan tenslotte ook gebruikt worden via de symbolische naam BUFFER, zie hiervoor hoofdstuk 11, symbolische adressen.

Het rekenorgaan van de SERA bestaat uit een aantal registers, waarvan 2 rechtstreeks programmatisch bereikt kunnen worden. Deze registers worden genoemd A en B accumulator.

Sommige opdrachten zouden tijdens de uitvoering aanleiding kunnen geven tot onopgemerkte fouten, zoals overloop, delen door 0, enz. Voor deze doeleinden is een speciaal alarmregister aanwezig, waarvan de stand door een speciale voorwaardelijke sprong kan worden afgevraagd. Die opdrachten, die aanleiding kunnen geven tot dergelijke fouten, beïnvloeden dit alarmregister. Indien de operatie goed verloopt, wordt het alarmregister in ieder geval afgezet, bij foutieve werking wordt het aangezet. Wil men het alarmregister dus gebruiken dan moet het worden afgevraagd vóór het door een volgende opdracht - die dit zou kunnen - weer wordt beïnvloed. Bij de bespreking van de verschillende opdrachten in volgende hoofdstukken zal expliciet worden aangegeven welke instructie het alarmregister kan aan- of afzetten.

## 2. TRANSPORTOPDRACHTEN

2.0. In dit en de volgende hoofdstukken betekent steeds (n) de inhoud van het geheugenwoord met adres n. Gelijke betekenissen hebben ook b.v. (A) en (B) voor de inhoud van accumulator A resp. B.

### 2.1. Transportopdrachten naar en van het gewone geheugen

HPA	n	Haal	(n)	Positief in accu	A	(n)	→	A
HNA	n	Haal	(n)	Negatief in accu	$\bar{A}$	-(n)	→	A
HPB	n	Haal	(n)	Positief in accu	B	(n)	→	B
HNB	n	Haal	(n)	Negatief in accu	$\bar{B}$	-(n)	→	B
BPA	n	Breng	Positief uit A	naar	$\bar{n}$	(A)	→	n
BNA	n	Breng	Negatief uit $\bar{A}$	naar	n	-(A)	→	n



BPB	n	Breng	Positief	uit B	naar n	(B) → n
BNB	n	Breng	Negatief	uit $\bar{B}$	naar n	-(B) → n
BSA	n	Breng	Schoon	uit A	naar n	(A) → n
				daarna		0 → A
BSB	n	Breng	Schoon	uit B	naar n	(B) → n
				daarna		0 → B

## 2.2 Transportopdrachten voor buffergeheugen

HAB n:m Haal uit buffer naar A en B van positie n t/m positie m.

Het resultaat van deze opdracht is, dat de hexaden vanaf plaats n t/m m in de buffer als een gesloten groep naar A en B samengebracht worden. A en B accumulatoren samen worden hier beschouwd als één lengte van 96 bits. De overgebrachte hexaden staan rechts aangesloten in AB, dus hexade m op de meest rechtse plaats in B. Totaal kunnen met een HAB opdracht maximaal 16 hexaden worden overgebracht. Is het aantal overgebrachte hexaden kleiner dan worden de overige plaatsen van A en B met nullen gevuld. Indien  $m-n+1$  groter is dan 16 dan wordt teruggeteld vanaf m, 16 hexaden naar AB gebracht. Er moet gelden  $\geq n$ ;  $m \leq 160$ ;  $n \leq 160$ . Indien aan deze eisen niet is voldaan, wordt het alarmregister gezet en de opdracht niet uitgevoerd.

BAB n:m Breng uit A en B naar buffer van positie n t/m positie m.

Dit is de tegenhanger van de vorige opdracht. Voorzover toepasselijk gelden dezelfde conventies. Daarbij geldt bovendien, indien  $m-n+1$  groter dan 16 dan worden 16 hexaden naar de buffer gebracht op de 16 posities teruggeteld vanaf m; in dit geval worden overige aangegeven posities in de buffer toch niet verstoord. De inhoud van AB blijft onaangetast. Er gelden dezelfde eisen als bij HAB waardoor het alarmregister wordt beïnvloed.

BUS n:m Maak buffer schoon vanaf positie n t/m positie m.

De buffer wordt in de aangegeven posities gevuld met de combinatie 101001 (41) die bij de ponskaart overeenkomt met een ongeponste kolom, op ponsband geen vertegenwoordiging heeft en dus ook niet geponst wordt. Tenslotte bij de regeldrukker, vertegenwoordigt deze combinatie een spatie. Ook hier gelden dezelfde eisen als bij HAB met alle gevolgen.



De opdrachten HAB, BAB, en BUS zijn in SERA de enige opdrachten van het 2e adrestype.

### 3. REKENOPDRACHTEN VOOR VASTE KOMMA

- OPA n Opttellen (n) in A  $(A) + (n) \longrightarrow A$   
 De optelling vindt decimaal plaats. De overdracht bij capaciteitsoverschrijding gaat verloren, maar wordt wel signaleerd via het alarmregister.
- AFA n Afttrekken (n) in A  $(A) - (n) \longrightarrow A$   
 zie opmerking bij OPA.
- OPB n Opttellen (n) in B  $(B) + (n) \longrightarrow B$   
 zie opmerking bij OPA
- AFB n Afttrekken (n) in B  $(B) - (n) \longrightarrow B$   
 zie opmerking bij OPA
- OAB n Opttellen in A en B  $(B) + (n) \longrightarrow B$   
 Een eventuele overdracht bij capaciteitsoverschrijding wordt direkt verwerkt in A.
- AAB n Afttrekken in A en B  $(B) - (n) \longrightarrow B$   
 zie opmerking bij OAB
- VMG n Vermenigvuldig (n) met (B)  
 Het produkt van dubbele lengte komt rechts aangesloten in A en B samen. De oude inhoud van A is van geen belang en gaat verloren. Tekens van het produkt wordt volgens normale regels teken van vermenigvuldiger maal teken van vermenigvuldiggetal. Dit geldt ook als een of beide factoren +0 of -0 zijn.
- DLN n Deling (AB) door (n)  
 Voor uitvoering van deze opdracht moet het deeltal als dubbele lengte getal rechts aangesloten in AB staan. Na uitvoering van de deling staat het quotient in B en de rest in A; in beiden rechts aangesloten. Het quotient heeft het correcte teken. Verder geldt  

$$\left| \text{rest} \right| < \left| \text{deler} \right|$$
 Tekens van de rest is altijd teken van deeltal. Voor het uitvoeren van de deling worden indien nodig de tekens van de beide delen van het deeltal gelijk gemaakt overeenkomstig de regels bij schuifopdrachten. Voor uitvoering van de deling moet gelden  $\left| \Delta \right| < \left| \text{deler} \right|$  - bovendien moet de deler ongelijk 0 zijn.



Indien aan deze eisen niet is voldaan wordt de deling niet uitgevoerd en het alarmregister gezet.

#### 4. SPRONGOPDRACHTEN

4.0 Zoals ook reeds vermeld in hoofdstuk 1.1, is het eerste bit van de tekentetrade maatgevend voor de rest bij de volgende voorwaardelijke sprongopdrachten.

##### 4.1 Absolute en voorwaardelijke sprongopdrachten

SAL n Spring altijd naar n

SPA n Spring naar n indien (A) positief of  $\pm 0$

SNA n Spring naar n indien (A) negatief

SPB n Spring naar n indien (B) positief of  $\pm 0$

SNB n Spring naar n indien (B) negatief

SPA en SPB gelden dus volledig voor het geval teken groter of gelijk aan nul.

SNA en SNB gelden uitsluitend voor het teken kleiner dan.

SOA n Spring indien (A) is gelijk  $\pm 0$

SOB n Spring indien (B) is gelijk  $\pm 0$

SAB n Spring indien (A) = (B)

Voor deze tekst zijn alle 48 bits van het woord van even grote betekenis. Vergelijking van (A) en (B) vindt hier plaats ongeacht de betekenis, die zij verder hebben. Voor deze tekst is dan ook +0 niet gelijk aan -0.

SIA n Spring indien alarmregister is gezet.

Door deze opdracht verandert de stand van het alarmregister zelf niet.

STP n SToP

Deze opdracht is bedoeld voor het tijdelijk ophouden van het programma b.v. voor het inleggen van een nieuwe band of nieuwe ponskaarten of voor het opzetten van een nieuwe magneetband. Na het indrukken van een bepaalde starttoets herneemt het programma zijn loop weer vanaf het aangegeven adres n.

##### 4.2 Subprogramma-opdracht

SSP n Spring naar subprogramma dat begint op adres n.

Alvorens de absolute sprong naar adres n wordt uitgevoerd wordt eerst het "terugkeeradres", d.w.z. het adres volgend op dat, waar de SSP opdracht staat, opgeborgen in een speciaal register. Dit register is bereikbaar onder de symbolische naam TERUG.



In verband met subprogramma's is nog het volgende van belang. Het speciale register TERUG heeft een volledige woordlengte van 48 bits waarin door SSP opdracht alleen het adresgedeelte en eventueel de kenmerketrade wordt ingevuld. Het laatste geval doet zich alleen voor bij een SSP sprong vanuit het dood geheugen of vanuit een speciaal register.

De in 5.1 genoemde absolute sprongopdracht SAL wordt in machine-code gerepresenteerd door de combinatie 00 in het operatiedeel. Hierdoor heeft de inhoud van het speciale register TERUG als instructie bekeken de betekenis van spronginstructie naar het terugkeeradres. Het einde van een subprogramma kan daardoor eenvoudig worden aangegeven door de instructie SAL TERUG. Deze instructie heeft zelf de betekenis spring naar speciale register TERUG. Daar staat dan de echte terugkeerinstructie. Door de speciale vorm van de terugkeerinstructie kan de inhoud van TERUG ook zonder meer in het subprogramma gebruikt worden om b.v. gegevens op te halen die een plaats hebben relatief t.o.v. de subprogramma sprong.

## 5. SCHUIFOPDRACHTEN EN LOGISCHE OPDRACHTEN

### 5.0 Decimale schuifopdrachten

Voor uitvoering van de volgende opdrachten wordt de inhoud van A en B beschouwd als een dubbele lengte decimaal getal. Zijn daarbij de tekens van A en B niet gelijk dan worden deze als volgt gelijk gemaakt. Indien  $(A) = +0$  en  $(B) \neq 0$ , dan wordt het teken van (A) gelijk aan het teken van (B) gemaakt. Indien  $(A) \neq 0$  dan wordt het teken van (B) gelijk aan dat van (A) gemaakt waarbij een overdracht van B naar A op de juiste manier wordt verwerkt. Indien zowel (A) als (B) gelijk  $+0$  dan worden (A) en (B) beiden  $+0$  gemaakt. Bij het schuiven zelf worden de tekentetraden niet meegeschoven.

ABL n Schuif A en B n decimale plaatsen naar links.

Het schuiven is niet rondgekoppeld. Rechts in B worden nullen behorende bij het teken bijgetrokken. De tetraden links uit A gaan verloren.

ABR n Schuif A en B n decimale plaatsen naar rechts.

Hiervoor een overeenkomstige opmerking als bij ABL. Bij deze opdrachten mag n ook 0 zijn. Er wordt dan niet geschoven maar wel worden zonodig de tekens gelijk gemaakt.

### 5.1 Binaire schuifopdrachten

Onder andere voor het werken met alfanumerieke informatie zijn volgende binaire schuifopdrachten aanwezig. Hierbij



worden A en B samen als een lengte van 96 bits beschouwd, die samen zowel naar links als naar rechts rondgeschoven kunnen worden, ongeacht de betekenis die de woorden in A en B hebben. Links uit A komt dus rechts in B en omgekeerd.

BSL n Binair schuiven naar links over n binaire plaatsen.

BSR n Binair schuiven naar rechts over n binaire plaatsen.

## 5.2 Logische bewerkingen

ENA n De logische bewerking "en" van (A) en (n) (A) n A

Deze bewerking geldt voor alle 48 bits. Indien van de sterfaciliteit gebruik gemaakt wordt, wordt het patroon van de 4 adrestetraden genomen en verder opgevuld met nullen.

OFA n De logische bewerking "of" van (A) met (n) (A) (n) A  
zie opmerking bij ENA

ENB n De logische bewerking "en" van (B) met (n)  
zie opmerking bij ENA

OFB n De logische bewerking van "of" van (B) met (n)  
zie opmerking bij ENA

## 6. MODIFICATIE-OPDRACHT

MOD n Modificeer de volgende opdracht voor de uitvoering met (n).

De modificatie strekt zich daarbij uit over het gehele woord, maar vindt plaats in de vorm van een decimale optelling.

## 7. IN- EN UITVOER-OPDRACHTEN

LSK Lees een kaart

De informatie van 80 kolommen van een ponskaart wordt overgebracht naar de posities 1 t/m 80 van de buffer. De samenhang tussen ponskaartcode en SERA hexade wordt gegeven door de tabel in hoofdstuk 12.

PSK Pons een kaart

Hiervoor gelden dezelfde conventies als voor LSK. LSK en PSK zijn adresloze operaties.

LSP n Lees een ponskaartn karakters en plaats die in posities 1 t/m n in de buffer.

Omzetting van ponscode naar SERA hexaden vindt weer plaats overeenkomstig de tabel in hoofdstuk 12. Aangezien hier is aangenomen dat op de 5-kanaalsponsband gewerkt wordt met zg. telexcode kan het voor-



komen dat het aantal SERA hexaden niet overeenkomt met het aantal symbolen op de ponsband b.v. door het optreden van letter/cijferwisselingen. Het adresgedeelte n heeft daarom betrekking op het aantal in te lezen SERA hexaden.

PSP n Pons in ponsband de eerste n symbolen uit de buffer.  
zie verder opmerking bij LSP

DRU Druk een regel op de regeldrukker.  
Afhankelijk van het type aangesloten regeldrukker wat betreft de regelbreedte wordt een regel van 120 of 160 symbolen afgedrukt. Dit vindt dan plaats overeenkomstig de eerste 120 of 160 posities in de buffer. Ook deze opdracht is adresloos.

DRN n Druk n nieuwe regels.  
Dit is voor het snel opschuiven van het papier in de regeldrukker.

## 8. KONVERTERINGSOPDRACHTEN

8.0 Door de verschillende voorstellingswijzen van informatie, die in SERA mogelijk zijn, komt het natuurlijk meermalen voor dat bepaalde informatie van de ene vorm in de andere moet worden omgezet. Hoewel dit in principe mogelijk is met de besproken opdrachten in voorgaande hoofdstukken, zou dit zeer omslachtig zijn. Daarom zijn er een aantal speciale konverteringsopdrachten mogelijk.

### 8.1 Konvertering numerieke en alfanumerieke vorm

KAG n Konverteer n hexaden van alfanumerieke vorm naar getalvorm.

Voor uitvoering van de opdracht moet een getal in hexadenvorm (alfanumeriek) rechts aangesloten in A en B samen staan. Na uitvoering staat het getal in tetradenvorm (getalvorm) rechts aangesloten in B. De oude inhoud van A is verloren gegaan en verder onbepaald. Eventuele hexaden die een spatie aangeven, worden genegeerd doch wel meegerekend voor het aantal n. Een aanwezige tekenhexade wordt op de juiste wijze verwerkt en meegerekend voor n. De konversie wordt dan van rechts naar links uitgevoerd. Indien geen tekenhexade aanwezig wordt het getal als positief aangenomen.

Indien  $n=0$  of  $n=16$  dan wordt n beschouwd als zijnde 16. De konversie-opdracht wordt beëindigd indien een der volgende gevallen zich voordoet.

1. n Hexaden zijn gekonverteerd.



2. Binnen de n hexaden wordt een tekenhexade ontmoet. Het gekonverteerde getal krijgt in dit geval nog wel het betrokken teken.
3. Binnen de n hexaden wordt een hexade ontmoet, die noch een getal, noch een teken, noch een spatie vertegenwoordigt.
4. Binnen de n hexaden zijn reeds 11 getalhexaden ontmoet. Verdere getalhexaden worden beschouwd als zijnde spaties.

KGA tsn Konverteer n hexaden van getalvorm naar alfanumerieke vorm.

n bepaalt het aantal gekonverteerde hexaden. Het getal n kan in het geschreven programma vooraf worden gegaan door een of beide der letters t en s. Door het invoerprogramma wordt het getal n van hooguit 2 cijfers tesamen met de indicaties t en s verwerkt tot één gegeven dat past in het adresgedeelte.

De betekenis van t en s is als volgt. Letter t afwezig betekent, het teken wordt onderdrukt. Letter t aanwezig betekent, het teken wordt gekonverteerd en als tekenhexade voor het getal gevoegd. Letter s aanwezig betekent, niet significante nullen worden gekonverteerd in spatiehexaden. Dit geldt voor alle posities. Is dus het te konverteren getal 0 dan ontstaat geen enkel echt cijfer. Letter s afwezig betekent, niet significante nullen worden ook omgezet in nul-hexaden. Voor zover toepasselijk gelden bij deze opdracht ook de konventies van de opdracht KAG.

Verder geldt hier speciaal indien het te konverteren getal in B niet voldoet aan de eisen voor een decimaal getal, dan wordt de opdracht niet uitgevoerd en het alarmregister gezet.

De (16-n) overblijvende hexaden in AB worden gelijk gemaakt aan spatie-hexaden.

Indien t aanwezig moet  $n \leq 12$ ; indien t afwezig moet  $n \leq 11$ ; is hieraan niet voldaan dan wordt n automatisch 12 resp. 11 aangenomen. Dit laatste is ook het geval als  $n=0$ .

## 9. MAGNEETBANDOPDRACHTEN

Voor het gebruik van magneetbanden is aangenomen dat SERA werkt als een moderne machine met mogelijkheden van multi-processing, waarbij directe magneetbandopdrachten geregeld worden door een centraal coördinatorprogramma. De in het programma gegeven magneetopdrachten zijn dan in feite pseudo-opdrachten voor dit coördinatorprogramma. Het voordeel is dat men in een eigen programma nu zonder meer een



aantal magneetopdrachten, zelfs voor hetzelfde apparaat, direct achter elkaar kan geven waarbij de werkelijke afloop van deze opdrachten geregeld wordt door het coördinatorprogramma. Daarnaast bestaat een speciale testopdracht om via het coördinatorprogramma het klaarkomen van bepaalde gegeven startopdrachten af te vragen. Bij de volgende aan te geven pseudo-opdrachten voor magneetbandgebruik verwijst het adresgedeelte steeds naar een zgn. verzorgingsblok. Op adres  $n$  en eventueel de volgende twee adressen moet dan de volgende informatie aanwezig zijn.

- ( $n$ ) = nummer van het te gebruiken magneetbandapparaat
- ( $n+1$ ) = beginadres in het geheugen van waaruit geschreven of waar naartoe gelezen moet worden.
- ( $n+2$ ) = aantal woorden dat getransporteerd moet worden.

Voor schrijven en lezen van magneetbanden bestaan de volgende opdrachten.

MVS  $n$  Magneetband vooruit schrijven.

Er wordt op de magneetband van apparatuurnummer ( $n$ ) een blok ter lengte van ( $n+2$ ) woorden geschreven vanuit het geheugen te beginnen op adres ( $n+1$ ). Het schrijven op de magneetband vindt plaats vanaf de positie waarin de magneetband op dat moment staat.

MVL  $n$  Magneetband vooruit lezen.

Dit geldt voor transport van magneetband naar geheugen met dezelfde conventies als MVS.

MTL  $n$  Magneetband terug lezen.

De magneetband van apparatuurnummer ( $n$ ) wordt in terugwaartse richting gelezen vanaf positie waarin hij staat. Daarbij worden ( $n+2$ ) woorden getransporteerd. In het geheugen wordt nu ook het aflopend adresnummer opgeborgen, daarbij is ( $n+1$ ) dan ook het eerste opbergadres, in dit geval dus het hoogste adres.

Aangezien men in een programma op een gegeven moment zeker moet zijn dat een eerder gegeven opdracht van bovenstaand type afgewerkt is bestaat de volgende testopdracht.

TST  $n$  Test de eerder gegeven startopdracht voor apparaatnummer ( $n$ ). Hier zijn ( $n+1$ ) en ( $n+2$ ) niet van belang.

Heeft men voor een apparaat achter elkaar meerdere startopdrachten gegeven, dan geldt deze TST-opdracht voor de eerstgegeven startopdracht voor dat apparaat, voor zover die nog niet door een TST-opdracht eerder



is afgevraagd. In het algemeen dient men dus iedere startopdracht van het type MVS, MVL of MTL af te sluiten met een TST-opdracht. Behalve deze startopdrachten bestaan nog enkele magneetbandopdrachten, waarvan het klaarkomen niet afgevraagd kan worden door een TST-opdracht. Dit is geen bezwaar omdat een daarop volgende opdracht MVS, MVL of MTL toch pas uitgevoerd wordt indien deze opdrachten klaar zijn.

MSM n Magneetband schoon maken.

Op de band wordt als het ware een stuk overgeslagen overeenkomende met een lengte van  $(n+2)$  woorden. Dit geldt voor apparaatnummer  $(n)$ . Hier is  $(n+1)$  niet van betekenis. Deze opdracht werkt alleen in voorwaartse bewegingsrichting van de band.

MTZ n Magneetband terugspoelen zonder blokkeren.

De magneetband van apparaatnummer  $(n)$  wordt teruggespoeld. Daarna kan de magneetband weer gebruikt worden.

MTM n Magneetband terugspoelen met blokkeren.

Zelfde als MTZ, maar nu is magneetbandapparaat na terugspoelen niet meer programmatisch bereikbaar. Pas na ontgrendelen met de hand door de operator kan betrokken apparaat weer gebruikt worden. Deze opdracht is dus voor het veilig stellen van een magneetband die pas met belangrijke informatie is beschreven.

## 10. INVOERPROGRAMMA

In voorgaande stukken is reeds enkele malen gesproken over het invoerprogramma. Dit invoerprogramma bestaat volledig uit SERA-opdrachten, waarbij het is geplaatst in het dood geheugen van de SERA. Voor zover het invoerprogramma werkruimte nodig heeft, waarin ook geschreven moet kunnen worden, is deze ruimte ontleend aan het levend geheugen. Wat betreft de beschrijving van dit invoerprogramma zullen we hier volstaan met het vermelden van de konventies, waaraan een geschreven en daarna geponst programma moet voldoen opdat het door het invoerprogramma gelezen kan worden.

Het invoerprogramma zorgt voor de omzetting van de mnemotechnische benaming van het operatiegedeelte van opdrachten in de interne SERA-code. Voor het adresgedeelte van een opdracht mag gebruik gemaakt worden van symbolische adressen. Nadere bijzonderheden over deze symbolische adressen worden vermeld in hoofdstuk 11.

Wordt nu in een instructie geen adres aangegeven dan vult het invoerprogramma hiervoor zelf het absolute adres 0 van het levend geheugen voor in.



Er bestaan enkele inzetaanwijzingen of directieven, waarmee bepaalde informatie aan het invoerprogramma kan worden gemeld. Deze hebben dezelfde vorm als SERA-instructies, maar komen natuurlijk niet als zodanig in het geheugen.

BGN n Begin in te zetten vanaf adres n.

Deze instructies volgend op BGN worden in het geheugen geplaatst te beginnen bij adres n. Voor n mag ook gebruik worden gemaakt van een symbolisch adres, mits dit reeds eerder is gedeclareerd.

Een speciale toepassing hiervan is b.v. het vrijhouden midden in een programma van een bepaalde werkruimte zoals:

LYST        |        BGN                    LYST + 100

Vanaf symbolisch adres LYST worden 100 plaatsen vrijgehouden (zie ook addenden bij symbolische adressen in hoofdstuk 11)

Het invoeren van een programma kan beëindigd worden met :

SRT n Start programma vanaf adres n.

Hierbij mag n ook een symbolisch adres zijn.

Er bestaan nog enkele directieven die samenhangen met de symbolische adressering en besproken worden in hoofdstuk 11. Instructies worden steeds regel voor regel geschreven. Bij gebruik van ponskaarten wordt één regel per kaart geponst. Bij gebruik van ponsband wordt het einde van iedere regel automatisch gemarkeerd door het symbool "terugloopwagen" nieuwe regel.

In plaats van een instructie mag een regel ook een geheel getal bevatten al of niet voorzien van teken. Dit getal wordt dan in normale decimale voorstelling opgeborgen. Zou een getal bestaan uit meer dan 11 cijfers dan worden alleen de eerste 11 cijfers van links naar rechts geassembleerd. Alfnumerieke informatie kan direkt ingevoerd worden door gebruik van aanhalingstekens. Per regel mogen max. 8 symbolen worden gebruikt. B.V. de tekst STICHTING STUDIECENTRUM kan in 3 SERA-woorden ondergebracht worden. als volgt :

"STICHTIN"  
"G STUDIE"  
" CENTRUM"

Tussen de aanhalingstekens hebben dus ook spaties hun betekenis. Gebruikt men minder dan 8 symbolen tussen de aanhalingstekens dan worden de symbolen rechts aangesloten in het woord opgeborgen, terwijl links met nullen wordt aangevuld.

Daardoor is dus "TEKST"                    hetzelfde als "000TEKST"



terwijl "TEKST " en " TEKST" betekenen het woord TEKST aangevuld met 3 spaties rechts resp. links. Voor het invoeren in binaire vorm kan gebruik gemaakt worden van max. 12 cijfers in het 16-tallig stelsel. Daarbij worden de cijfers 10, 11, 12, 13, 14 en 15 gegeven door resp. A, B, C, D, E en F. Ter onderscheid met instructies wordt een dergelijk getal tussen ronde haakjes ( ) geplaatst. Niet significanten nullen mogen weggelaten worden b.v.

(73CFA2)  
(FFFF)  
(3F)

Het invoerprogramma verwerkt verder de reeds genoemde ster \* tussen operatie en adres voor zgn. adresloze operanden. Bij de bufferopdrachten wordt de dubbel punt in rekening gebracht als scheiding tussen de twee adressen. Adressen van bufferopdrachten zijn altijd absoluut.

## 11. SYMBOLISCHE ADRESSEN

In plaats van absolute adressen mogen ook symbolische adressen gebruikt worden. Deze beginnen altijd met een letter en bestaan verder uit een combinatie van letters en cijfers. Alleen de eerste 8 symbolen worden gebruikt voor indentificatie. Een symbolisch adres aan het begin van een regel betekent een declaratie waardoor die naam aan dat adres wordt toegekend. Men kan ook relatief adresseren t.o.v. een symbolisch adres. Daartoe schrijft men als adres een naam dan een plus- of minteken, daarna een geheel getal, b.v.

of SAL TABEL + 14  
SAL TABEL - 43

Verwerking van deze addenden vindt in het invoerprogramma plaats door gewoon decimaal optellen en aftrekken. Zou daarvoor een negatief adres ontstaan, dan wordt dit gerealiseerd door een eenheid te "lenen" van het operatiegedeelte. Dit geval kan zich voordoen, indien het een instructie betreft, die later tijdens de uitvoering van het programma gemodificeerd wordt met een positief adres. Addenden zijn ook toegelaten achter absolute adressen. Dit gecombineerd met de regel dat het absolute adres 0 mag worden weggelaten, laat de volgende constructie toe.

HPA - 67

Teneinde in verschillende delen van een programma dezelfde namen voor verschillende toepassingen te kunnen gebruiken kunnen declaraties van namen weer ongedaan worden gemaakt. Dit is speciaal van belang bij gebruik van subprogramma's. Hiervoor bestaat het volgende directief.

VRy n

Hier is n een symbolisch adres, dat reeds in een statisch



(d.w.z. in de volgorde van het geschreven programma) voorafgaand deel van het programma is gedeclareerd. Voor het volgende noemen wij een programmadeel gelegen tussen een gedeclareerde naam en het directief VRY met dezelfde naam een blok. Door het directief VRY worden alle in dat blok gedeclareerde namen lokaal voor dit blok en hebben buiten dit blok hun betekenis verloren. Dit geldt zowel voor de programmadelen voor als achter het blok. Een programma kan dus bestaan uit een aantal achter elkaar liggende blokken. Dit noemen we parallel blokken, b.v.

```

A      VRY    A
B      VRY    B
C      VRY    C

```

Dit zijn 3 parallelblokken A, B en C.

Blokken mogen elkaar ook omvatten, b.v.

```

A
B      VRY    B
      VRY    A

```

Nu is blok B een binnenblok van blok A.

Symbolische adressen, die in een bepaald blok wel worden gebruikt maar niet gedeclareerd, zijn automatisch lokaal voor het omringde blok. Indien ook daarin niet gedeclareerd, dan tevens lokaal voor het omvattende blok, enz. Tenslotte wordt het gehele programma, ook al is dit niet afgesloten, met het directief VRY als een allesomvattende blok beschouwd. Wel gebruikte, maar nergens gedeclareerde namen zijn dus automatisch lokaal voor het hele programma. Ten aanzien van dergelijke namen geeft het invoerprogramma 2 mogelijkheden. Bij de 1e mogelijkheid worden aan deze namen na het directief SRT alsnog vrije adressen in de overblijvende geheugenruimte toegewezen alvorens het programma te starten. Bovendien kan op verzoek een gedrukte lijst van deze namen met toegewezen adressen worden verstrekt. Bij de 2e mogelijkheid is aangenomen, dat alle gebruikte namen in het programma door juiste declaratie reeds zijn toegewezen. In dat geval betekent een niet gedeclareerde naam dus een programma fout, er wordt dan alleen een lijst van deze namen gegeven en het programma wordt niet gestart. Door het directief VRY worden alle gedeclareerde namen in het blok weer vrij gemaakt. Tenslotte een opmerking t.a.v. vaste symbolische namen die in voorgaande hoofdstukken zijn genoemd als :

```

ACCU-A
ACCU-B
TERUG

```

Deze symbolische adressen mogen gebruikt worden zonder declaratie en hebben dan de in de vorige hoofdstukken aangegeven betekenis.



SERIE-A

Naam  
Adres  
Woonplaats  
Kursistennr.

Voor de onderstaande vragen dient U het rondje op te zullen van de antwoorden die volgens U goed zijn. Wij wijzen U er echter bij voorbaat op dat bij een vraag meerdere antwoorden en zelfs alle antwoorden goed kunnen zijn. Het tegenovergestelde kan zich ook voordoen. M.a.w. er kan op een bepaalde vraag ook geen enkel antwoord goed zijn.

Na het oplossen van de vragen dient U de pagina's van Serie-A in te sturen aan E.C.S, Postbus 2 Heerlen.

Veel succes.

- 
- Vraag 1.    0    Gegevens, indien op gewoon papier geschreven, zijn als invoer voor de computer onbegrijpelijk.
- 0    Een SERA programma, dat in ponskaarten is geponst, wordt door de SERA computer zonder meer geaksepteerd en uitgevoerd.
- 0    De meeste computer-programma's kunnen, zonder dat gegevens-invoer noodzakelijk is, uitgevoerd worden.
- Vraag 2.    0    De SERA maakt gebruik van ponskaarten.
- 0    De SERA maakt gebruik van magneetband.
- 0    De SERA maakt gebruik van magnetische schijven.
- 0    De SERA maakt gebruik van ponsband.
- 0    De SERA maakt gebruik van optisch schrift.
- Vraag 3.    0    Een ponskaart kan per kolom slechts één karakter weergeven.
- 0    Een ponskaart kan per kolom 2 numerieke cijfers weergeven.
- 0    Een ponskaart kan geen speciale tekens en/of leestekens weergeven.



SERIE-A

Naam

Woonplaats

- Vraag 4. 0 De bovenrand van een ponskaart kan worden bedrukt met de tekst, die in de kaart is geponst.  
0 De 0-rij in een ponskaart wordt zowel als zoneponsing als numerieke ponsing gebruikt.  
0 In een normale ponskaart moeten steeds 80 kolommen worden geponst.  
0 Na het leesbevel voor een normale ponskaart worden er steeds 80 kolommen ingelezen.
- Vraag 5. 0 De sneldrukker is een in- en uitvoerapparaat.  
0 De sneldrukker is alleen een uitvoerapparaat.  
0 Op een sneldrukker worden onder besturing van de centrale eenheid alle tekens achtereenvolgens afgedrukt.
- Vraag 6. 0 Magneetband is de meest gebruikte en snelste vorm van in- en uitvoer.  
0 De informatie op een magneetband is niet te zien.  
0 Het schrijven en lezen van bits op een magneetband berust op de inductie-theorie uit de elektriciteitsleer.
- Vraag 7. 0 Een SERA hexade wordt op de magneetband in lengte richting geschreven.  
0 Een SERA hexade wordt op de magneetband in dwars richting geschreven.  
0 De SERA machine maakt gebruik van een 8-kanaals magneetband.  
0 De SERA machine maakt gebruik van een 6-kanaals magneetband.
- Vraag 8. 0 Het taktspoor op een magneetband zorgt voor het mechanische transport van de band.  
0 Het taktspoor geeft aan wanneer er een kolom met informatie onder de lezer komt.  
0 De kanaal-pariteitsbits dienen om de kolom-pariteitbits te controleren.
- Vraag 9. 0 De dichtheid van een SERA magneetband bedraagt 22 kolommen per mm.  
0 8 kolommen van een magneetband kunnen een SERA woord bevatten.



SERIE-A

Naam

Woonplaats

- 0 De snelheid van een SERA magneetband is ongeveer 2,5 m per seconde.
- Vraag 10. Om te lezen van of te schrijven op de magneetband wordt de band langs de schrijf- leeskop getrokken via :
- 0 De losse haspel.
- 0 De vaste haspel.
- 0 Een aandrijfmechanisme.
- 0 Een elektromotor via koppelingen.
- Vraag 11. De magneetband hangt in het magneetbandapparaat in lussen om :
- 0 Grotere blokken naar de band te kunnen schrijven.
- 0 Het starten en stoppen van de band in zeer korte tijd te laten gebeuren.
- 0 Het breken van de band bij vertragen of versnellen tegen te gaan.
- 0 Om het einde van de band vast te kunnen stellen.
- Vraag 12. 0 Blokhiaten bevatten pariteitsbits.
- 0 Tussen iedere twee SERA woorden op de band bevindt zich een blokhiaat.
- 0 Tussen 2 informatieblokken op de magneetband bevindt zich een blokhiaat.
- 0 De lengte van een blokhiaat is afhankelijk van de tijd, die nodig is om de magneetband op volle snelheid te brengen en om hem af te remmen.
- Vraag 13. Een voordeel van het gebruik van een grote blok-lengte is :
- 0 De magneetband wordt doelmatig gebruikt.
- 0 Geringer tijdverlies i.v.m. minder stoppen en starten.
- 0 Dat bij fout lezen of schrijven niet zoveel terug gespoeld hoeft te worden.
- Vraag 14. 0 De SERA "buffer" maakt deel uit van het SERA geheugen.
- 0 De SERA "buffer" is een apart stuk geheugen.
- 0 De SERA "buffer" is opgebouwd uit 160 posities



SERIE-A

Naam

Woonplaats

en heeft dezelfde structuur als het SERA geheugen.

- Vraag 15. 0 De SERA "buffer" bevat 160 posities om de inhoud van 2 ponskaarten in te kunnen lezen.
- 0 In de SERA "buffer" wordt de informatie opgeslagen in de speciale kaartcode.
- 0 In de SERA "buffer" wordt de informatie opgeslagen in de SERA hexadencode.
- 0 De SERA "buffer" dient zowel voor invoer als voor uitvoer.
- Vraag 16. 0 De LSK-opdracht brengt de inhoud van een ponskaart over naar de eerste 80 posities van de SERA "buffer".
- 0 De PSK-opdracht ponst de informatie uit de SERA "buffer" van positie 81 t/m 160 in een ponskaart.
- 0 De DRN n -opdracht drukt de informatie uit de SERA "buffer" van positie 1 t/m 120 n keer op de sneldrukker af.
- Vraag 17. 0 Vanuit een ponskaart kan de SERA machine nooit de informatie rechtstreeks in het geheugen plaatsen.
- 0 Vanuit de "buffer" kan informatie-overdracht rechtstreeks plaatsvinden naar het geheugen.
- 0 Vanuit de "buffer" kan informatie-overdracht rechtstreeks plaatsvinden naar de accumulatoren.
- Vraag 18. 0 KAG betekent, konverteer van hexade naar tetra-de.
- 0 KGA betekent, konverteer van tetra-de naar hexade.
- 0 KGA zonder ts betekent, geen teken maar wel de niet significante nullen omzetten.
- Vraag 19. 0 Alle informatie in de buffer heeft de hexadenvorm.
- 0 Een getal neemt in hexadenvorm meer plaats in dan in tetradenvorm.
- 0 Konversie van hexaden naar tetraden is alleen noodzakelijk voor getallen waarmee gerekend moet worden.



SERIE-A

Naam

Woonplaats

- Vraag 20. In accu-B staat het getal -00063591068 in tetradenvorm.  
Na uitvoering van de opdracht KGA tb staat er in accu-B (hexaal) :
- 0 §§ - 91068
  - 0 §§ 591068
  - 0 00591068
  - 0 - 0591068
- Vraag 21. Wanneer men met HAB 1 positie uit de buffer leest en men weet dat dit een cijfer is :
- 0 Dan is de opdracht KAG overbodig.
  - 0 Dan is de opdracht KAG noodzakelijk.
  - 0 Dan is de opdracht KGA noodzakelijk.
- Vraag 22. 0 Een KGA opdracht kan max. 16 karakters (accu A + accu B) van tetradenvorm naar hexadenvorm converteren.
- 0 Alle niet via konversie tot stand gekomen hexaden in de accu's A en B na een KGA opdracht worden gelijk gemaakt aan de spatie-hexade.
  - 0 Alle niet via konversie tot stand gekomen hexaden in de accu's A en B na een KGA opdracht worden gelijk gemaakt aan de nulhexade.
- Vraag 23. Een KAG n instructie wordt tijdens de uitvoering beëindigd :
- 0 Alleen indien de n de hexade een tekenhexade is.
  - 0 Indien n hexaden zijn gekonverteerd.
  - 0 Als er een tekenhexade wordt ontmoet.
  - 0 Indien er reeds 12 getalhexaden zijn gekonverteerd.
- Vraag 24. Enkele mogelijkheden om basis gegevens direkt in de computer te lezen zijn :
- 0 Mark sensing.
  - 0 Magnetische inkt.
  - 0 Ponskaart.
  - 0 Optisch schrift.
  - 0 Magneetband.



SERIE-B

B.1

Onderstaande vragen en opgaven dient U allen voor Uzelf uit te werken. De opgaven voorzien van drie kruisjes (+++) dient U echter uit te werken op het bijgevoegde uitwerkpapier en in te sturen ter beoordeling aan E.C.S. Postbus 2 Heerlen.

---

- Vraag 1.a. Geef in een eenvoudige tekening de technische details van een ponskaart.  
b. Noem 2 kaartlees principes.
- Vraag 2.a. Welke ponsbandbreedtes zijn in gebruik?  
+++  
b. Wat is de betekenis van de zgn. letter- en cijfer symbolen?  
c. Waarom komen deze alleen bij een 5-kanaals ponsband voor?  
d. Noem 2 ponsband-lees principes.
- Vraag 3. Noem enkele technische gegevens van de sneldrukker t.a.v. regelbreedte, snelheid, afdrukmechanisme.
- Vraag 4.a. Geef schematisch in een tekening weer hoe een compleet SERA-woord op een magneetband wordt geplaatst.  
b. Waarvoor dient het taktspoor?  
c. Waarom wordt er gewerkt met een dubbele parateitscontrole.
- Vraag 5. Noem en beschrijf de SERA ponskaart-opdrachten.  
+++
- Vraag 6. Noem en beschrijf de SERA sneldrukker-opdrachten.
- Vraag 7.a. Noem en beschrijf de SERA ponsband-opdrachten, die betrekking hebben op de "buffer"  
b. In welke gevallen geeft men aan ponsband de voorkeur boven ponskaarten?
- Vraag 8.a. Noem en beschrijf de SERA buffer-opdrachten.  
b. Wat valt U op aan de operand?  
c. Hoe wordt deze operand in een SERA-instructie opgenomen?
- Vraag 9. Beschouw de accu's als totaal 16 hexaden. Hoe zien na HAB de hexadenplaatsen er uit die niet nodig zijn voor het resultaat van deze opdracht?  
+++



Vraag 10.a. Wanneer heeft men BUS nodig?

- b. Stel dat DRN niet bestaat. Hoe kan men deze opdracht dan in een stukje programma nabootsen m.b.v. DRU?

Vraag 11. Cijfers (getallen) komen meestal in alfanumerieke vorm aan bij de computer. Voor berekeningen dienen zij in numerieke vorm te staan. Bij de output (uitvoer) ligt dit juist andersom. Welke twee vorm opdrachten (konversie-opdrachten) bestaan hiervoor in de SERA?  
Geef hun volledige mogelijkheden.

Vraag 12. Gegeven twee kaarten met alfanumerieke tekens  
+++ in kolom 20 t/m 35, 40 t/m 55 en 60 t/m 75.

- a. Hoeveel woorden zijn min. nodig om de totale tekst op te bergen?  
b. Geef het programmadeel dat deze kaarten leest en opbergt op adres 1000 en volgende.

Vraag 13. Als vraag 12, doch nu bevatten de kolommen 20 t/m 31, 40 t/m 51 en 60 t/m 71 een getal met teken. Breng deze getallen over naar 2000 en volgende.

- a. Met teken en met significante nullen.  
b. Zonder teken en met significante nullen.  
c. Als a, zonder significante nullen.  
d. Als b, zonder significante nullen.

Dit geeft dus 4 programmadeeltjes.

Vraag 14. In accu B staat het volgende tetra-de-getal  
+++ -00000086420.

Dit getal wordt op verschillende manieren gekonverteerd. Het resultaat van een spatie-hexade mag als  $\$$  worden weergegeven. Hoe luiden de resultaten voor de volgende konversies :

- |           |             |
|-----------|-------------|
| a. KGA    | f. KGA t8   |
| b. KGA 2  | g. KGA t2   |
| c. KGA 6  | h. KGA ts3  |
| d. KGA s5 | i. KGA ts5  |
| e. KGA s8 | j. KGA ts10 |

Vraag 15. Gevraagd een programmadeel om een paar kaarten te verwerken. De sluitkaart heeft in kolom 85



een nul en dan moet worden doorgesprongen naar de eerste instructie van de rest van het programma. In kolom 5 t/m 21 van iedere kaart staat een naam, in kolom 25 t/m 29 een salaris in centen. Druk de namen af op de sneldrukker en bepaal tegelijkertijd de som der salarissen in een hulpadres van het programma.

Vraag 16.  
+++

Gegeven een pak ponskaarten. Het aantal is onbekend doch kleiner dan 10.000. In kolom 1:6 van iedere kaart is een getal geponst. Elk getal is groter dan nul behalve dat van de laatste kaart, dat gelijk is aan nul.

Van deze kaarten moet worden bepaald :

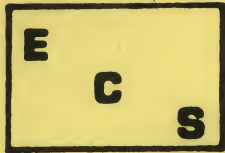
- a. Het aantal incl. de laatste.
- b. De som van alle getallen die bestaan uit resp. 1, 2, 3, 4, 5 en 6 cijfers.

Als alle kaarten zijn verwerkt moet worden gesprongen naar een programma dat begint op adres 5000. Bedenk een oplossing en geef deze weer in een stroomschema, waaruit dan duidelijk blijkt hoe het probleem luidde.

Past elk resultaat onder elk denkbare situatie in het woord?



# e u r o p e a n   c o m p u t e r   s c h o o l



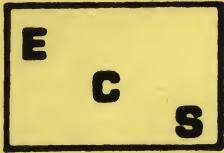
Naam :  
Adres :  
Woonplaats :  
Kursistennr. :

PASSAGE 4-6. HEERLEN

U I T W E R K P A P I E R



# eu r o p e a n      c o m p u t e r      s c h o o l



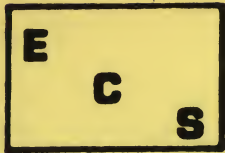
Naam :  
Adres :  
Woonplaats :  
Kursistennr. :

PASSAGE 4-6. HEERLEN

U I T W E R K P A P I E R



# European computer school



Naam :  
Adres :  
Woonplaats :  
Kursistennr. :

PASSAGE 4-6. HEERLEN

U I T W E R K P A P I E R